

PONTIFÍCIA UNIVERSIDADE CATÓLICA DO RIO GRANDE DO SUL
FACULDADE DE INFORMÁTICA
PROGRAMA DE PÓS-GRADUAÇÃO EM CIÊNCIA DA COMPUTAÇÃO

**UM ESTUDO EMPÍRICO SOBRE
MÁQUINAS DE TRADUÇÃO EM TEMPO
REAL PARA EQUIPES DISTRIBUÍDAS DE
DESENVOLVIMENTO DE SOFTWARE**

JOÃO HENRIQUE STOCKER PINTO

Dissertação apresentada como
requisito parcial à obtenção do grau de
Mestre em Ciência da Computação na
Pontifícia Universidade Católica do Rio
Grande do Sul.

Orientador: Prof. Dr. Rafael Prikladnicki

Porto Alegre

2016

Ficha Catalográfica

P659e Pinto, João Henrique Stocker

Um Estudo Empírico Sobre Máquinas de Tradução em Tempo Real para Equipes Distribuídas de Desenvolvimento de Software / João Henrique Stocker Pinto .

– 2016.

82 f.

Dissertação (Mestrado) – Programa de Pós-Graduação em Ciência da Computação, PUCRS.

Orientador: Prof. Dr. Rafael Prikladnicki.

1. Máquinas de Tradução. 2. Máquinas de Tradução de Voz. 3. Desenvolvimento Distribuído de Software. 4. Reconhecimento de Voz. 5. Síntese de Voz. I. Prikladnicki, Rafael. II. Título.

Elaborada pelo Sistema de Geração Automática de Ficha Catalográfica da PUCRS com os dados fornecidos pelo(a) autor(a).

DEDICATÓRIA

Para minha namorada Marina, minha mãe Karla e meu pai João Américo.

AGRADECIMENTOS

Ao longo dos últimos dois anos passei por diversos desafios pessoais e profissionais e o resultado de um destes desafios agora está pronto. Este trabalho, porém, não seria possível sem a ajuda e o apoio de várias pessoas que estiveram presentes neste período.

Sendo assim, gostaria de agradecer à minha família, pelo suporte e pelas sábias palavras e conversas que tivemos nos momentos em que eu estava cansado de tudo.

Ao meu amigo e professor Rafael Prikladnicki, por ter me aceitado no mestrado, pelos conselhos ao longo destes últimos 2 anos e pela sua forma de conduzir o trabalho – me mostrando um caminho e não me puxando por ele.

Ao professor Marcelo Hideki Yamaguti, por confiar em meu trabalho desde o tempo da faculdade até a indicação para o mestrado.

À professora Sabrina Marczak, por me apoiar desde o primeiro momento em que pensei sobre o mestrado e por estar sempre presente quando precisei.

Ao professor Michael da Costa Móra, pelos momentos de convivência, ensinamentos e nas tomadas de decisão em assuntos da Aceleradora.

Ao meu amigo Alejandro Olchik, por ter me ensinado tanto sobre Metodologias Ágeis e pelas diversas sessões em que pude participar.

À PUCRS, à Faculdade de Informática e ao TECNOPUC, por dispor de uma excelente infra-estrutura e localização.

À ThoughtWorks, pela bolsa de estudo integral que me foi proporcionada.

Aos professores Fabio Calefato e Filippo Lanubile, por me ajudarem com os experimentos que fiz e por terem me recebido de forma tão cordial enquanto estive em Turin, na Itália em 2014.

Aos meus colegas de mestrado e principalmente para o meu amigo Bernardo Estácio, valeu pelos momentos de estudo e pela parceria até aqui.

UM ESTUDO EMPÍRICO SOBRE MÁQUINAS DE TRADUÇÃO EM TEMPO REAL PARA EQUIPES DISTRIBUÍDAS DE DESENVOLVIMENTO DE SOFTWARE

RESUMO

O desenvolvimento distribuído de software está cada vez mais presente na cultura das empresas de T.I. O número de empresas que distribui os integrantes das suas equipes buscando reduzir custos de mão de obra, aumentar a qualidade dos produtos e também aumentar a produtividade cresce anualmente. Este cenário, porém, demanda uma grande cooperação entre pessoas que, em muitos casos, não dominam o mesmo idioma. Máquinas de tradução em tempo real são uma alternativa a este cenário, realizando a tradução simultânea de um idioma para outro. Nesta dissertação de mestrado investiga-se, através de estudos empíricos, a tradução em tempo real em equipes de desenvolvimento distribuído de software utilizando-se a voz. Para isso, apresenta-se uma revisão histórica do surgimento das ferramentas de reconhecimento, tradução e síntese de voz até o seu estado atual, além de abordar características técnicas das mesmas. A base empírica da pesquisa conta ainda com dois experimentos realizados em parceria com a Universidade Aldo Moro, de Bari, na Itália, utilizando as tecnologias disponíveis no mercado, além do desenvolvimento de dois protótipos que fazem a integração entre reconhecedores de voz, máquinas de tradução e sintetizadores de voz. A pesquisa contribui no sentido de mostrar quais tecnologias atualmente disponíveis para a comunicação entre equipes distribuídas que não dominam o mesmo idioma estão próximas de serem efetivas e se elas podem ser usadas nas atividades diárias em equipes de desenvolvimento de software. Além disso a pesquisa aponta compatibilidades entre ferramentas e um possível caminho a seguir para integrar reconhecedores de voz, máquinas de tradução e sintetizadores de voz.

Palavras chaves: Máquinas de Tradução de Voz, Desenvolvimento Distribuído de Software, Reconhecimento de Voz, Síntese de Voz.

AN EMPIRICAL STUDY OF REAL-TIME SPEECH TRANSLATION TOOLS FOR MULTILINGUAL DISTRIBUTED SOFTWARE DEVELOPMENT TEAMS

ABSTRACT

Distributed Software Development is increasingly present into the culture of information technology companies. The number of companies that spread its teams trying to reduce costs, improve products quality and improve productivity increases every year. This scenario, however, demands a huge cooperation between people that, in many cases, do not master the same language. A Speech Translation System is an alternative to this scenario, simultaneously translating from a language to another. This master thesis presents an empirical study, which consists of the historical review of the rise of recognition tools, translation and speech synthesis to its current state, as well as addressing technical characteristics of the same. The empirical research base has two experiments conducted in partnership with the University Aldo Moro of Bari, in Italy, using part of the tools available in the market and in development of two prototypes that make the integration of speech recognition, machine translation and speech synthesis to facilitate communication between distributed teams of software projects. The research contributes in order to show that the currently available technologies for communication between distributed teams that don't dominate the same language are close to be really effective and if they can be used in daily activities in software development teams. In addition to compatibility between tools, this research tries to point which the way forward to integrate voice Recognizers, Machine Translation and Speech Synthesis.

Keywords: Voice Translation Machines, Voice Translation, Distributed Software Development, Speech Recognition, Speech Synthesis.

LISTA DE FIGURAS

Figura 1 – Hype Cicle da Gartner para tecnologias emergentes de 2013.....	17
Figura 2 – Hype Cicle da Gartner para tecnologias emergentes de 2014.....	17
Figura 3 – Hype Cicle da Gartner para tecnologias emergentes de 2015.....	18
Figura 4 – Principais razões envolvidas no DDS.....	24
Figura 5 – Máquina de Artsrouni.....	29
Figura 6 – Máquina de Troyanskii.....	30
Figura 7 – Sintetizador de Voz com Fole de Kempelen.....	33
Figura 8 – Desenho da Pesquisa.....	35
Figura 9 – Uso do <i>Google Translate</i> no Dispositivo <i>Mobile</i>	40
Figura 10 – Google Web Speech API.....	44
Figura 11 – Permissão para acesso ao microfone.....	47
Figura 12 – Print Screen do protótipo Web.....	48
Figura 13 – Lista de amigos do <i>Skype</i> funcionando no protótipo.....	50
Figura 14 – Mensagem sendo recebido pelo próprio cliente do <i>Skype</i>	50
Figura 15 – Etapas de uma Máquina de Tradução de Voz.....	52
Figura 16 – Aplicações e Restrições das Tecnologias de Reconhecimento de Voz.....	55
Figura 17 – Aplicações e Restrições das Máquinas de Tradução.....	56
Figura 18 – Aplicações e Restrições dos Sintetizadores.....	56

LISTA DE TABELAS

Tabela 1 – Combinação dos Pares de Idiomas.....	38
Tabela 2 – Alguns Exemplos de Sentenças Utilizadas no Experimento 1.....	39
Tabela 3 – Precisão do Reconhecimento de Voz e da Tradução.....	41

LISTA DE GRÁFICOS

Gráfico 1 – Precisão do Reconhecimento de Voz.....	41
Gráfico 2 – Precisão das Traduções.....	42
Gráfico 3 – Taxa de Acertos no teste preliminar 1 para IT-EN e EN-IT.....	53
Gráfico 4 – Taxa de Acertos no teste preliminar 2 para IT-EN e EN-IT.....	54

LISTA DE ABREVISTURAS E SIGLAS

ALPAC – *Automatic Language Processing Advisory Committee*

API – *Application Programming Interface*

ASCII – *American Standard Code for Information Interchange*

ASR – *Automatic Speech Recognition*

DARPA – *Defense Advanced Research Project Agency*

DDS – *Desenvolvimento Distribuído de Software*

ES – *Engenharia de Software*

HMM – *Hidden Markov Model*

IBM – *International Business Machines*

MT – *Máquina de Tradução*

PUCRS – *Pontifícia Universidade Católica do Rio Grande do Sul*

RE – *Requirements Engineering*

SW – *Software*

TI – *Tecnologia da Informação*

TTS – *Text-To-Speech*

SUMÁRIO

LISTA DE FIGURAS.....	9
LISTA DE TABELAS.....	10
LISTA DE GRÁFICOS.....	11
LISTA DE ABREVIATURAS E SIGLAS.....	12
1. INTRODUÇÃO.....	17
1.1. OBJETIVOS.....	19
1.2. MOTIVAÇÃO.....	20
1.3. ORGANIZAÇÃO DO VOLUME.....	21
2. REFERENCIAL TEÓRICO.....	22
2.1. DESENVOLVIMENTO DISTRIBUÍDO DE SOFTWARE.....	22
2.2. RECONHECIMENTO DE VOZ.....	24
2.2.1. DEFINIÇÃO E CONCEITOS.....	24
2.2.2. PANORAMA ATUAL.....	25
2.3. MÁQUINAS DE TRADUÇÃO.....	26
2.3.1. HISTÓRIA DAS MÁQUINAS DE TRADUÇÃO E EVOLUÇÃO.....	26
2.3.2. TIPOS DE ABORDAGENS EM MÁQUINAS DE TRADUÇÃO.....	30
2.4. SÍNTESE DE VOZ.....	31
2.4.1. ORIGENS.....	31
3. METODOLOGIA DE PESQUISA.....	33
4. EXPERIMENTOS E RESULTADOS DOS ESTUDOS.....	34
4.1. EXPERIMENTO 1 – MOBILE.....	35
4.1.1. ANÁLISE CRÍTICA.....	41
4.2. EXPERIMENTO 2 – WEB.....	41
4.2.1. ANÁLISE CRÍTICA.....	43

4.3.	PROTÓTIPO 1 – WEB.....	43
4.4.	PROTÓTIPO 2 – JAVA DESKTOP.....	46
4.5.	ESTUDO DE VIABILIDADE TÉCNICA – SKYPE TRANSLATOR.....	48
4.6.	MAPA DAS TECNOLOGIAS.....	51
5.	CONSIDERAÇÕES FINAIS.....	54
5.1.	CONTRIBUIÇÕES DA PESQUISA.....	54
5.2.	LIMITAÇÕES DA PESQUISA.....	55
6.	PESQUISAS FUTURAS.....	56
6.1.	REFLEXÃO FINAL.....	56
7.	REFERÊNCIAS BIBLIOGRÁFICAS.....	57
	APÊNDICE – PROTOCOLO DE ESTUDO – EXPERIMENTO 1.....	70
	APÊNDICE – PROTOCOLO DE ESTUDO – EXPERIMENTO 2.....	75

1. INTRODUÇÃO

“Ter um sonho grande dá o mesmo trabalho que ter um sonho pequeno.”

Jorge Paulo Lemann

O desenvolvimento de software e principalmente a engenharia de requisitos (*requirements Engineering - RE*) são atividades que demandam uma intensa comunicação [DD07]. Em projetos globais, esta comunicação é acentuada [NUS00]. Com a crescente globalização das empresas de TI, a dificuldade em elicitar requisitos – que antes era local – passou a uma escala continental e conseqüentemente multilíngue [DD03].

Aspectos como fuso-horário, política, economia e diferenças culturais estão cada vez mais presentes em *RE* assim como a cooperação entre as equipes distribuídas. Esta cooperação normalmente acontece através da comunicação, que em razão de fatores históricos é feita no idioma inglês [CAR01] [LB09]. Em muitos destes projetos o idioma se torna um dos fatores primordiais para o sucesso ou fracasso das equipes e do produto final. Além disso, é sabido que elicitar requisitos para projetos de grande escala não é trabalho para uma pessoa sozinha. Esta tarefa requer um grupo de pessoas, coordenação entre estas pessoas, estimar prioridades e tempo, priorizar regras de negócio e tudo isto envolvendo programadores, analistas, testadores, gerentes de projeto, entre outros [YH06].

Países que possuem um mercado de TI ativo, como a Irlanda, Filipinas, Índia, Singapura, os Estados Unidos e alguns países da América Latina demandam de profissionais que possam se entender de forma concisa. No Brasil, por exemplo, estima-se que apenas 5.4% da população saiba falar inglês, ou seja, 10.2 milhões de habitantes somente. A Índia conta com um índice também baixo, de 8.2% da população, mas acaba sendo maior em números absolutos. Com um crescimento do mercado de TI com uma taxa de 6.5% ao ano desde 2005, o Brasil se apresenta como uma clara oportunidade para o desenvolvimento de software com equipes distribuídas [ITB10] [CAL11].

Dentro deste contexto, este trabalho visa estudar como máquinas de tradução em tempo real baseadas em voz podem ser utilizadas no contexto do Desenvolvimento Distribuído de Software.

Especificamente, deseja-se também entender como a tecnologia atual pode apoiar equipes de desenvolvimento de software brasileiras no contexto do desenvolvimento de software global, mesmo que a equipe não domine o mesmo idioma das equipes distribuídas em outros países, ou que o inglês não seja o idioma principal para comunicação. Esta pesquisa é a continuação de dois outros estudos: um deles iniciado por Calefato [CAL10] em 2010 e o outro por Duarte [DUA14] [DUA14b] em 2014. Em seu trabalho Calefato investigou a adoção de máquinas de tradução em serviços de *chat* em tempo real, onde os *stakeholders* estavam remotamente negociando requisitos de *software*. Naquela ocasião ele usou duas máquinas de tradução para rodar seus experimentos, o *Google Translate*, que também será utilizado em nosso trabalho e a *Apertium*, uma máquina de tradução *open source* disponível para o público em geral. O reconhecimento de voz, porém, não foi explorado, abrindo um caminho para que esta área fosse investigada em um novo estudo conduzido por Duarte [DUA14b].

No trabalho de Duarte [DUA14b], a pesquisa teve seu foco não apenas nas máquinas de tradução, mas também na investigação da existência de estudos que envolvessem a tradução automática de voz para o idioma português brasileiro, ficando, porém, apenas na parte teórica. O estudo não explorou a avaliação das tecnologias e seu potencial de uso em equipes de desenvolvimento de software.

Em relação às tecnologias de tradução em tempo real, o *Hype Cycle* (um estudo anual sobre tendências, maturidade e adoção de determinadas tecnologias ou aplicações, publicado pelo Gartner [GAR15]) apresentava entre 2013 e 2014 uma crescente tendência em tradução de voz-para-voz e no reconhecimento de voz [DUA14b]. Assim, o Gartner entende que estas tecnologias poderiam estar entre as mais inovadoras dos próximos 5 a 10 anos e 2 a 5 anos, respectivamente, como podemos notar em destaque nas figuras 1 e 2 abaixo.

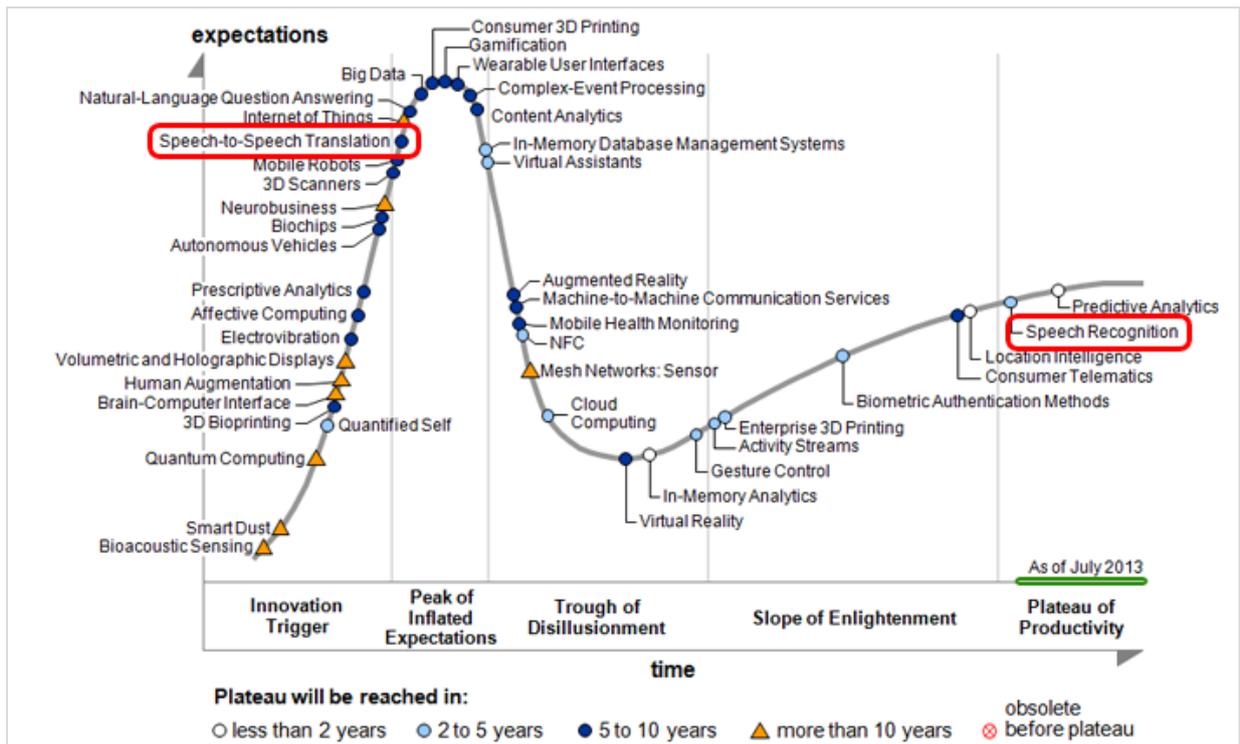


Figura 1 – Hype Cycle da Gartner para tecnologias emergentes de 2013.

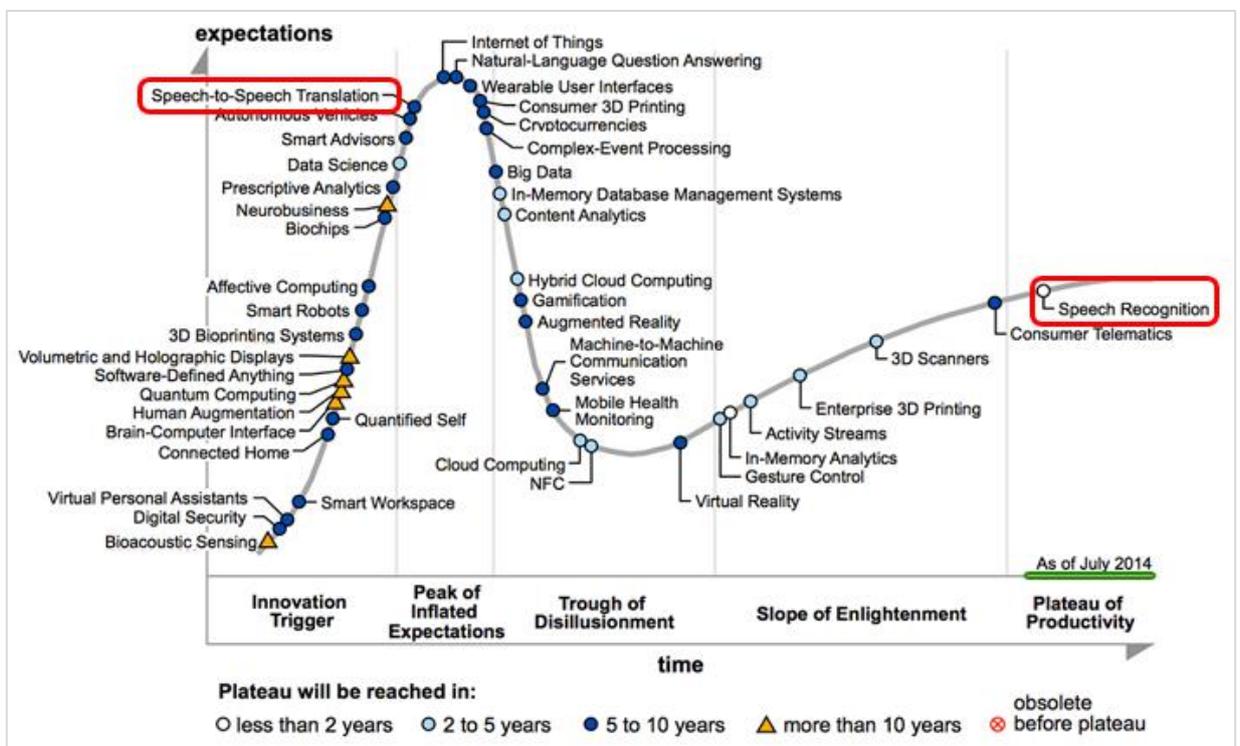


Figura 2 – Hype Cycle da Gartner para tecnologias emergentes de 2014.

Na figura 3, a seguir, o *Hype Cycle* deste ano mostra que o reconhecimento de voz não figura mais no gráfico, tendo passado pelo “*Plateau of Productivity*”. Isto comprova que a tendência analisada nos anos de 2013 e 2014 estava certa, pois tecnologias que se encontram ou já passaram pelo *Plateau* começam a ser adotadas pelas principais companhias, seus critérios de avaliação estão bem definidos e a aplicabilidade no mercado começa a gerar resultados. Ao compararmos a posição da tradução de voz entre 2013 e 2014 com a expectativa deste ano, podemos notar que a sua posição começou a descer a curva do “*Peak of Inflated Expectations*”. Isto significa que neste ponto um grande número de publicações relacionadas ao assunto se mostra bem sucedida, da mesma forma que o fracasso de muitas publicações também acontece, pois é neste momento onde as empresas decidem por apostar nesta tecnologia ou então se manter estagnada [GAR15].

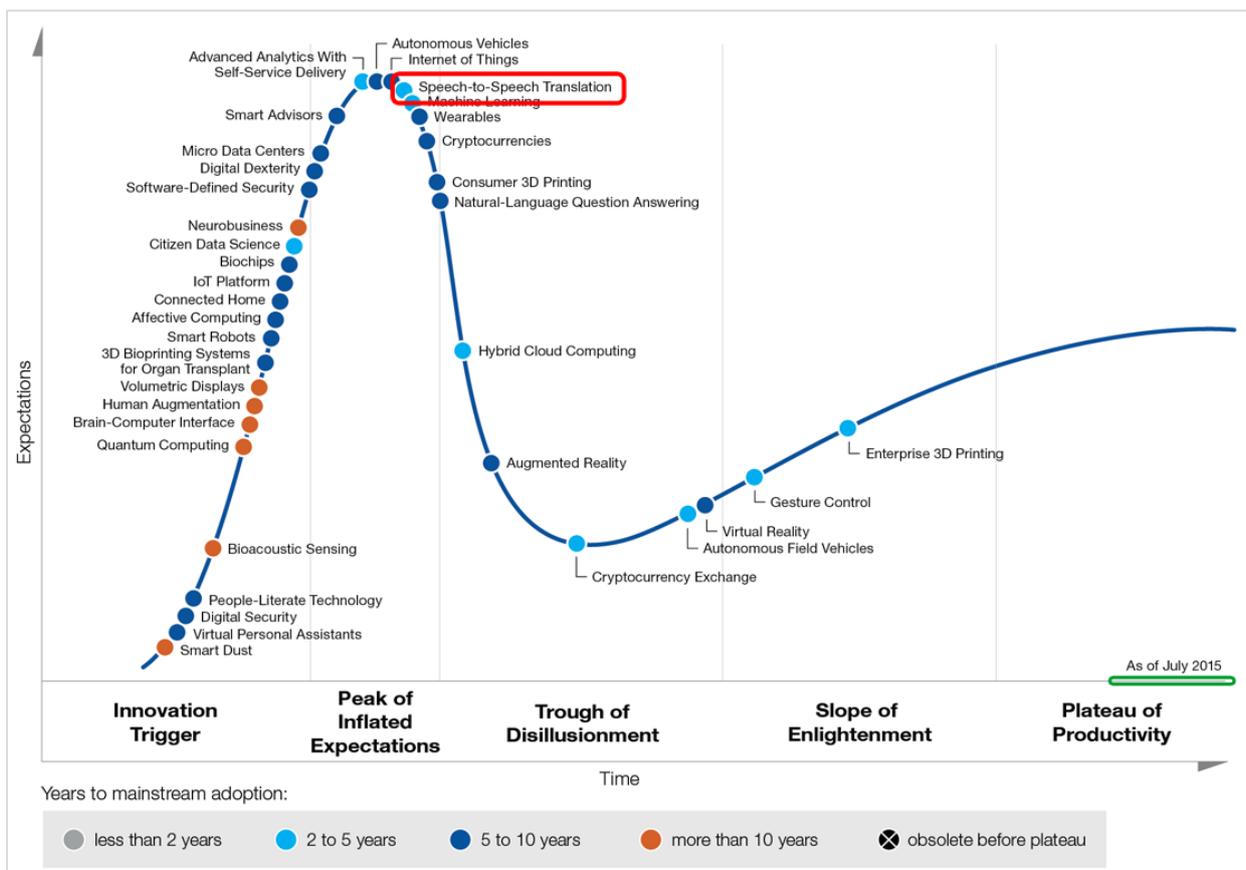


Figura 3 – Hype Cycle da Gartner para tecnologias emergentes de 2015.

Desta forma, a questão de pesquisa que norteou este estudo é a seguinte:

QP - Como a tecnologia de tradução em tempo real baseada em voz pode apoiar equipes distribuídas de desenvolvimento de software em escala global que não dominam o mesmo idioma?

Este trabalho tem como base empírica três experimentos realizados em parceria com os pesquisadores Filippo Lanubile e Fabio Calefato, na Universidade Aldo Moro, de Bari, na Itália. Junto destes experimentos foram desenvolvidos dois protótipos, um *web* cuja linguagem predominante é *Javascript* e outro *desktop* onde a principal linguagem utilizada foi Java. Estes experimentos serviram para automatizar o processo de reconhecimento de voz, tradução e síntese e os seus resultados serão apresentados ao longo do trabalho.

1.1. OBJETIVOS

Nesta seção são apresentados tanto o objetivo geral quanto os objetivos específicos.

O **objetivo geral** é verificar como a tradução em tempo real pode ajudar na comunicação e no entendimento entre equipes distribuídas de desenvolvimento distribuído de software em escala global, focando principalmente no processo de reconhecimento de voz e nas máquinas de tradução.

De forma a complementar o objetivo geral proposto, apresentam-se os seguintes **objetivos específicos**:

- Revisar e apresentar um estudo sobre a história e o surgimento do reconhecimento de voz, máquinas de tradução e síntese de voz;
- Apresentar os conceitos teóricos sobre Desenvolvimento Distribuído de Software;

- Identificar como máquinas de tradução podem ser utilizadas em projetos de DDS onde os integrantes dos times não falam o mesmo idioma;
- Analisar o processo de reconhecimento de voz e seu estado da arte;
- Testar o comportamento e o nível de confiabilidade das tecnologias atuais de reconhecimento de voz;
- Verificar a viabilidade técnica de se unir, utilizando a linguagem de programação Java, o reconhecimento de voz com máquinas de tradução e posterior síntese de voz.

1.2. MOTIVAÇÃO

Nos últimos anos pode-se perceber um grande avanço na globalização dos negócios e na área de desenvolvimento de software não é diferente. O software se tornou um componente estratégico para diversas áreas de negócio e ao mesmo tempo em que a engenharia de software (ES) se desenvolve, é necessário também manter a alta qualidade do SW em si [PFL04]. Os mercados nacionais da área de engenharia de software tem se transformado em mercados globais, criando novas formas de competição e cooperação que vão além das fronteiras físicas dos países [HER01]. Paralelo a isto, tem se tornado cada vez mais trabalhoso e menos competitivo desenvolver *softwares* no mesmo espaço físico, na mesma organização e até no mesmo país.

Os avanços nas economias globais, a diversa gama de ferramentas de comunicação e a pressão por opções mais baratas tem incentivado um grande investimento no Desenvolvimento Distribuído de Software. Estes investimentos tem criado um movimento de transformação de um mercado local para mercados globais, desenvolvendo um processo que facilita a criação de novas formas de colaboração e competição [DAM06].

Além destes fatores, diversos autores e pesquisadores tem destacado os desafios e as práticas usadas no DDS [CAR99] [CAR02] [EVA04] [HER07] [LOP05] [MOC01] e diversas empresas de desenvolvimento de software tem

lançado produtos que visam aproximar pessoas através da comunicação [SKY14b] [NUA16] [GOO16c].

Juntando isto com alguns dos principais fatores que contribuíram para acelerar o surgimento do DDS, como a necessidade de recursos globais para serem utilizados a qualquer hora, a vantagem de estar perto de mercados locais, incluindo o conhecimento dos clientes e as condições locais para explorar oportunidades de mercado e a grande pressão para o desenvolvimento *time-to-market* [CAR99] [CAR055], estudos e pesquisas sobre a utilização de máquinas de tradução em tempo real para o desenvolvimento de software tornam-se de grande relevância para esta área.

1.3. ORGANIZAÇÃO DO VOLUME

Este volume está organizado em 7 capítulos. Na sequência, no capítulo 2 é apresentado o referencial teórico desta pesquisa, mostrando os principais aspectos e fatores que envolvem o desenvolvimento distribuído de software. Também são apresentados os principais conceitos, o histórico e as características que tornam os processos de reconhecimento de voz, máquinas de tradução e síntese de voz um assunto de grande importância para a área da computação em geral.

No capítulo 3, apresenta-se a metodologia de pesquisa, descrevendo cada uma das etapas do estudo e como foi traçado o plano lógico de cada passo deste estudo.

No capítulo 4 são apresentados os experimentos, estudos de viabilidade tecnológica e protótipos que foram desenvolvidos com a finalidade de avaliar as tecnologias disponíveis usadas neste trabalho, bem como o detalhamento, a elaboração e os resultados obtidos em cada um dos estudos – *Mobile*, *Web* e *Skype* – sendo apresentadas comparações entre ferramentas, suas limitações e fatores que foram considerados impactantes no sucesso ou fracasso das tecnologias usadas.

As considerações finais sobre o tema, as contribuições para nossa área e as limitações do estudo de forma geral são apresentadas no capítulo 5. Pesquisas futuras e a reflexão final são apresentadas no capítulo 6 e, por fim, o capítulo 7 apresenta as referências bibliográficas.

2. REFERENCIAL TEÓRICO

“No que diz respeito ao empenho, ao compromisso, ao esforço, à dedicação, não existe meio termo. Ou você faz uma coisa bem feita ou não faz.”

Ayrton Senna

Este capítulo apresenta a base teórica dos principais conceitos abordados durante o plano de pesquisa. Na seção 2.1, serão apresentados os aspectos relativos ao desenvolvimento distribuído de software. Na seção 2.2 o reconhecimento de voz é apresentado, seus conceitos, sua definição e o panorama atual são detalhados. Na seção 2.3 falaremos sobre as máquinas de tradução, desde sua origem conhecida, passando pelas peculiaridades das linguagens e as dificuldades de montar uma gramática até o seu uso no dia a dia. A síntese de voz, outro componente importante para o trabalho é apresentada na seção 2.4, também falando de sua origem, características e principais usos.

2.1. DESENVOLVIMENTO DISTRIBUÍDO DE SOFTWARE

O gerenciamento de grandes projetos de desenvolvimento de software tem se tornado uma tarefa cada vez mais difícil, seja pelo grande número de necessidades dos clientes, pelas diferenças entre as equipes ou pelo próprio fuso-horário. Cada vez mais podemos notar que a distribuição física das equipes está se tornando um padrão em TI, pois as grandes empresas estão cada vez mais espalhadas pelo mundo com o nível de globalização atual. Estes são alguns dos fatores que caracterizam o Desenvolvimento Distribuído de Software (DDS) [PRI02]. Além dos fatores citados anteriormente, as empresas buscam constantemente ganhos de produtividade, melhorias nos processos, melhorias na qualidade dos produtos finais e diminuição de riscos, tornando os requisitos para a criação destes softwares cada vez mais complexos e sofisticados.

O tradicional trabalho feito pelas equipes locais, permitindo um grande fluxo de informações entre seus componentes está dando lugar a esta nova forma de desenvolver software e o ambiente de trabalho está cada vez mais adaptado a esta nova realidade, onde podemos perceber no nosso dia a dia o avanço tecnológico para diminuir estas distâncias criadas pela distribuição das equipes. Dentre estes diversos avanços, podemos destacar o acesso à informação em alta velocidade e as diversas ferramentas de colaborações existentes [SIQ05].

Sendo assim, desenvolver software de forma distribuída é uma atividade ainda mais complexa. Se notarmos algumas das principais categorias sugeridas por [Prikladnicki 2008] no desenvolvimento software, encontramos tecnologia, processos e pessoas, onde a tecnologia diz respeito às diversas ferramentas e aparatos tecnológicos que podem ser utilizados como apoio ou base para o desenvolvimento do projeto, o processo diz respeito à forma como o projeto será elaborado e as pessoas dizem respeito as características que afetam diretamente os recursos humanos envolvidos no desenvolvimento de software.

Estas 3 categorias aumentam de forma considerável quando consideramos o processo de desenvolvimento distribuído de software, onde as principais razões foram subdivididas por Prikladnicki da seguinte forma:



Figura 4 – Principais razões envolvidas no DDS.

Nos capítulos que seguem apresentaremos os conceitos e características das tecnologias que tangem um dos campos que mais contribuem para a sinergia cultural, que é a comunicação, na forma de reconhecimento de voz, máquinas de tradução e síntese de voz.

2.2. RECONHECIMENTO DE VOZ

A conversação é o principal meio de comunicação entre as pessoas. Por razões que vão desde curiosidade tecnológica até a realização de experimentos mecânicos, o desejo de automatizar tarefas requer a iteração entre humanos e máquinas. Uma das grandes atrações das últimas cinco décadas é como fazer uma máquina capaz de reconhecer a voz [JUA04].

Sendo assim, o reconhecimento de voz é o primeiro dos três recursos que será apresentado, pois ele é o ponto inicial para que uma máquina de tradução em tempo real possa realizar sua função, sendo utilizado com a finalidade de ser o ponto de entrada da máquina e a sua saída sendo o ponto de entrada para o tradutor propriamente dito.

2.2.1. DEFINIÇÃO E CONCEITOS

Automatic Speech Recognition (ASR) ou Reconhecimento Automático de Voz é definido como a transcrição de palavras faladas em texto [HBR14]. Produzir a transcrição de uma fala contínua e sem pausas como numa conversa é desafiador. Pesquisas nessa área datam do início de 1970 [RED76] porém, desde a década de 30, quando Homer Dudley, da *Bell Laboratories* propôs um modelo de sistema para análise de voz, o assunto tem evoluído.

Na década de 70, a maioria dos pesquisadores eram de universidades ou do governo americano, quando foi fundada a DARPA – *Defense Advanced Research Project Agency* ou Agência de Defesa para Projetos de Pesquisa Avançados. Esta agência foi responsável pela grande evolução e criação de

algumas das formas para o reconhecimento de voz: seu programa tinha como objetivo criar um sistema de computador que fosse capaz de reconhecer discursos contínuos [DRA14]. Dentro das principais tecnologias criadas pela DARPA, estava o *Harpy*, um reconhecedor que tinha um vocabulário de 1.011 palavras e uma precisão considerada boa. A principal contribuição do *Harpy* para a área foi o conceito da busca de palavras através de grafos, onde a língua para o reconhecimento da voz era representada como uma rede léxica de representações de palavras, com regras para produção sintática.

Já na década de 80, uma outra metodologia para o reconhecimento de voz começou a ficar popular: o *hidden Markov model* (HMM) ou modelo oculto de Markov. O HMM é um modelo estatístico que tem parâmetros desconhecidos. O desafio é buscar os parâmetros ocultos a partir dos parâmetros observados. Este processo é usado em ASRs, pois pode ser aplicado para o reconhecimento de padrões, assim como nós temos na fala [JUA04].

2.2.2. PANORAMA ATUAL

Atualmente, existem várias empresas que fazem pesquisas na área e possuem produtos comerciais, como a LumenVox, a Nuance, a própria Microsoft com o seu Speech SDK, fazendo parte do *.NET Framework*, o Google com o *Web Speech API* e o Apple com a Siri [CAL14]. Segundo Melanie Pinola da *PCWorld*, em 2001 o reconhecimento de voz alcançou 80% de precisão, porém estes resultados eram todos baseados em regras de estatísticas, até que em 2010 o Google adicionou um sistema customizado de reconhecimento de voz nos aparelhos com Android, isto fez com que as pesquisas por voz ficassem cada vez mais apuradas, criando-se uma gigantesca base de dados que em 2011 incorporava por volta de 230 bilhões de palavras das *queries* feitas pelos usuários [PM11].

Um das inovações que temos atualmente é o sistema de reconhecimento da voz da Apple, Siri, que é baseado em processamento na nuvem. Ela filtra o que conhece sobre o usuário para poder dar uma resposta

ainda mais apurada. Segundo David Daw, também da *PCWorld*, o reconhecimento de voz passou de utilitário para o entretenimento, onde este tipo de sistema permitirá ao usuário o controle total de seus equipamentos [DD11].

2.3. MÁQUINAS DE TRADUÇÃO

Máquinas de tradução podem ser definidas como o uso de um computador para traduzir de uma determinada língua, a língua origem para uma língua destino [JM08]. Nesta seção será apresentada uma parte da história, sua evolução ao longo do tempo, os principais conceitos sobre máquinas de tradução e métodos utilizados nas traduções.

Além disto, neste trabalho as máquinas de tradução foram utilizadas como o segundo recurso para que os protótipos de tradução em tempo real realizassem sua função, recebendo como entrada as transcrições dos reconhecedores de voz e fornecendo como saída os textos traduzidos para os sintetizadores de voz. A seguir serão apresentados alguns dos principais fatores históricos dessa tecnologia.

2.3.1. HISTÓRIA DAS MÁQUINAS DE TRADUÇÃO E EVOLUÇÃO

Quando as máquinas de tradução começaram a ficar populares no final dos anos 50 e início dos anos 60, as pessoas começaram a procurar por antecedentes históricos. Alguns traçam as origens das MTs no século 17, com as ideias de Descartes, Leibniz e outros filósofos e estudiosos da área da filosofia e universo, porém propostas de máquinas consultando dicionários e fazendo traduções não se materializaram até o início do século 20, com um maior desenvolvimento da tecnologia.

O primeiro modelo conhecido de uma máquina de escrever e tradutora foi demonstrada pelo estoniano A.Vakher (Vaxer) em fevereiro de 1924. Os relatos

dessa demonstração constam nos anais de uma conferência sobre máquinas de tradução ocorrida em 1962 em Tallinn, Estônia [HUT04] [HUT97].

Além desse relato, podem existir outras propostas para máquinas de tradução não reconhecidas, porém o primeiro registro legítimo conhecido data de 1933, quando na França em 22 de julho e na Rússia em 5 de setembro, Georges Artsrouni e Petr Trojanskii, respectivamente, aplicaram para duas patentes de dispositivos eletromecânicos usados para tradução de dicionários [HUT93].

A patente dada a Georges B. Artsrouni era de um dispositivo de aplicações gerais, que servia para guardar e imprimir informações. Artsrouni sugeriu que o dispositivo serviria para a produção automatizada dos horários dos trens, de listas telefônicas, de códigos de telégrafos comerciais e de extratos bancários. Por fim, sustentou-se que a máquina também poderia ser usada como dispositivo de tradução. Em seus primeiros relatos sobre o aparelho, em 1933, Artsrouni viu que uma de suas principais aplicações seria para a tradução mecânica de palavras. Ele afirmou que "o cérebro" da máquina poderia ser adaptado para a "tradução de uma língua estrangeira para umas das outras três línguas gravadas na máquina" e que "a máquina poderia ser ilimitada dependendo dos dicionários utilizados".

Como um dicionário mecânico, o "cérebro" era composto por quatro componentes básicos, uma memória de palavras disponível em 4 línguas, um dispositivo de entrada que consistia de um teclado, um dispositivo de pesquisa e um componente para a saída das informações, ativado pela cabeça de leitura. Os componentes funcionavam com a ajuda de um motor e a estrutura que dava suporte aos mecanismos era uma caixa retangular medindo aproximadamente 25x40x21 cm [CM60] [HUT04].

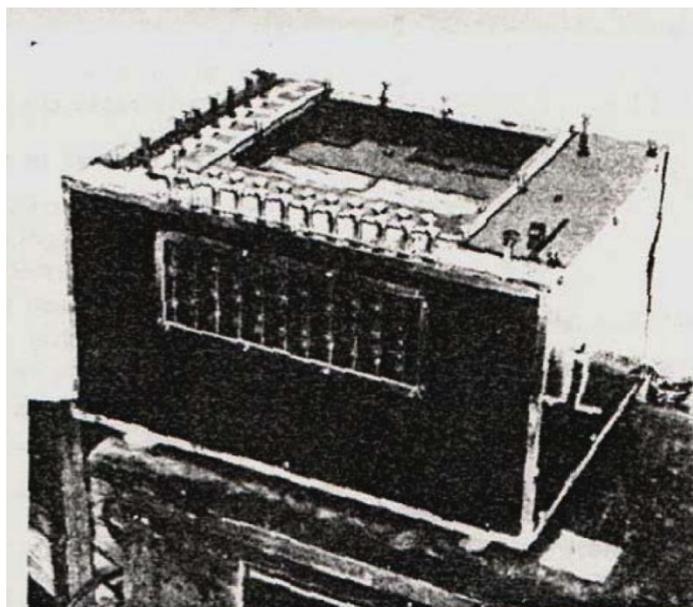


Figura 5 – Máquina de Artsrouni.

Outro precursor dos trabalhos em máquinas de tradução foi o russo Petr Petrovich Troyanskii. Nascido em janeiro de 1894 ao sul dos Montes Urais, era de uma família de operários de ferrovias. Estudou na Universidade de São Petersburgo e no “*Institute of Red Professors*” do partido Bolchevique. Assim como Artsrouni, Troyanskii em 1933 conseguiu uma patente para sua máquina de tradução. O dispositivo foi certificado como sendo uma “máquina para seleção de palavras digitadas e tradução para outras línguas simultaneamente” [HUT00].

A máquina de Troyanskii consistia de uma mesa suavemente inclinada, na qual uma cinta perfurada podia se movimentar em diferentes direções por uma abertura. Esta cinta era na verdade um dicionário de palavras com entrada para seis línguas diferentes postas em colunas paralelas. As traduções eram feitas por um operador da máquina e este tinha que localizar a palavra na língua origem, mover a cinta até achar a palavra na língua destino e então digitar. Depois de fazer este processo, a palavra resultante era passada para uma fita de saída e o processo se repetia até que uma frase fosse criada. Por fim, um revisor interpretava a sentença e produzia o resultado final [HUT04].

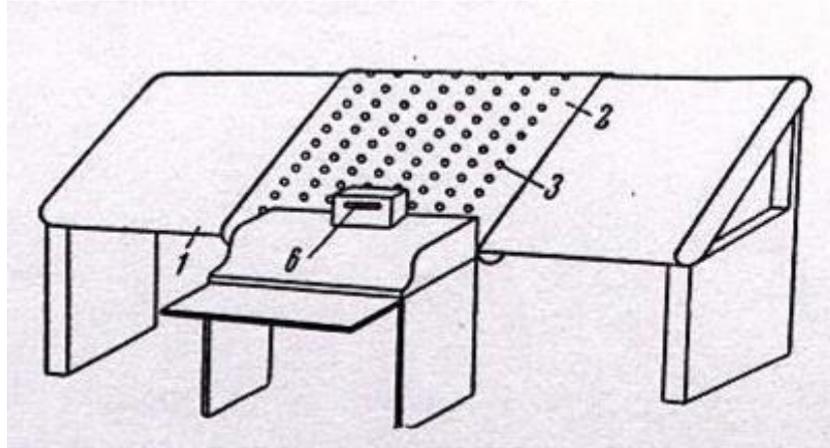


Figura 6 – Máquina de Troyanskii.

Até este ponto na história, os dispositivos existentes eram bastante manuais e a ideia de dispositivo digital para traduções surgiu pela primeira vez com Warren Weaver, entre 1947 e 1949, com a publicação de um memorando [WEA49] [HUT97b]. Este memorando é considerado como o lançamento dos Estados Unidos na pesquisa sobre máquinas de tradução. Nele Weaver relata a importância das diferenças culturais entre as línguas e a importância dos computadores durante o tempo de guerra, onde a velocidade, capacidade e a lógica seriam fatores para se analisar em máquinas de tradução nos anos seguintes. Além disso, Weaver cita alguns pontos que são relevantes até hoje, como significado e contexto, linguagem e lógica, tradução e criptografia e os idiomas e suas variações [WEA49].

Após a entrada americana nas pesquisas sobre MTs, uma equipe que consistia de membros da IBM e da Universidade de Georgetown desenharam seu equipamento para traduções. Em janeiro de 1954 esta equipe fez a primeira demonstração americana de uma tradução, sendo relatada no jornal *The New York Times* como “Russo é transformado em Inglês por um rápido tradutor elétrico”. Apesar do grande feito, o experimento era de pequena escala, consistindo de apenas 250 vocabulários e de 6 regras [HUT04b]. Após estes experimentos, o governo americano passou a patrocinar a busca por métodos mais precisos e passou o foco das suas preocupações na falta de progressos que a tecnologia disponível apresentava. Em 1964 o governo americano criou o ALPAC – *Automatic Language Processing Advisory Committee*, um comitê que tinha como objetivo medir a evolução na linguística computacional em geral e

particularmente a evolução das máquinas de tradução. Em 1966 este comitê apresentou um relatório onde seus membros se mostravam céticos em relação às máquinas de tradução e sugeriam que as pesquisas deveriam ser interrompidas, porém após estes fatos, novas frentes de pesquisa surgiram não somente nos Estados Unidos, mas em todo mundo.

As décadas de 80 e 90 foram o momento do surgimento de uma grande variedade de máquinas de tradução, devido a popularização dos *mainframes* e posterior dos microcomputadores. Várias frentes para a tradução surgiram como o Logos (inglês-alemão e alemão-inglês), o sistema da *Pan American Health Organization* (espanhol-inglês e inglês-espanhol) e alguns outros sistemas para a tradução entre o japonês e o inglês [HUT05]. Inovações e resultados em experimentos foram publicados pela IBM, métodos usando uma abordagem com exemplos foram criados e iniciou-se a mistura entre as diferentes abordagens. Além disso as pesquisas que eram puramente teóricas passaram para a prática. Com o surgimento da internet, em meados de 1990, sites começaram a lançar serviços de tradução, como o *Alta Vista* e o seu *Babel Fish*, tradutor de textos ou páginas da internet [HIS10].

Atualmente, com a grande variedade de sistemas de tradução disponível, os principais são o *Bing Translator*, da Microsoft e o *Google Translate* do Google, este último atendendo a aproximadamente 200 milhões de usuários por dia, segundo dados da *CNET* de 2013 [CNE13] e também contando com aproximadamente 500 milhões de usuários ativos por mês, fazendo com que o tradutor atinja a marca de mais de 1 bilhão de traduções mensais [GOO15a] [GOO16a].

2.3.2. TIPOS DE ABORDAGENS EM MÁQUINAS DE TRADUÇÃO

Os sistemas por trás das máquinas de tradução podem ser classificados em duas grandes categorias, a *corpus-based* e a *rule-based*.

Na abordagem *corpus-based*, grandes coleções de textos armazenados em bases de dados são comparados com o texto ou a frase que o usuário está

querendo traduzir. Este estilo de tradução não tem a direta intervenção humana, é feita através de um motor de busca que aprende a fazer a melhor tradução. Embora este método seja mais barato, nem sempre é possível ter uma quantidade suficientemente grande de dados em todas as línguas desejadas, o que torna este estilo dependente destas bases.

Na abordagem *rule-based* o conhecimento usado se dá na forma de regras explicitamente codificadas por especialistas nas linguagens. Estes especialistas são responsáveis por cuidar das diferentes características existentes entre cada idioma, como estrutura léxica, morfologia e a transferência de regras entre as linguagens. Diferentemente da abordagem *corpus-based*, a dependência humana neste estilo é fundamental [CAL14].

2.4. SÍNTESE DE VOZ

Text-To-Speech (TTS) ou texto-para-voz é o processo de converter uma determinada entrada em forma de texto para uma saída em forma de som inteligível. Este processo é composto por duas etapas fundamentais: a análise e a síntese. Na análise, um segmento de texto serve como base para se montar a estrutura linguística e este resultado é transformado para sua forma acústica correspondente, ou seja, o som resultante, na síntese [SJ05] [ALL90].

A síntese de voz, por sua vez, foi o terceiro recurso explorado neste trabalho, usando como entrada as saídas fornecidas pelas máquinas de tradução e por fim disponibilizando aos usuários a síntese de voz em algum dos idiomas selecionados. A seguir serão apresentados alguns dos principais registros das origens dessa tecnologia.

2.4.1. ORIGENS

Os primeiros registros de um sistema capaz de sintetizar voz datam do século 18, quando em 1779, na Rússia, Ch. G. Kratzenstein construiu um modelo

mecânico capaz de reproduzir algumas vogais. Kratzenstein era um professor de fisiologia e ele construiu este equipamento usando a ressonância de tubos conectados por foles. Em 1791, Wolfgang von Kempelen conseguiu pela primeira vez obter sucesso em sintetizar a fala de forma conectada, utilizando um sistema pneumático para expelir o ar de um balão e reproduzir palavras do início ao fim [MAN10].

A máquina criada por Kempelen consistia de um fole que simulava os pulmões de uma pessoa e de aberturas que liberavam o ar para sintetizar o som. Conforme o usuário fizesse movimentos para abrir ou fechar estas aberturas, a pressão do ar que passava pela máquina produzia diferentes sons, mais anasalados, vogais e consoantes [TH00].

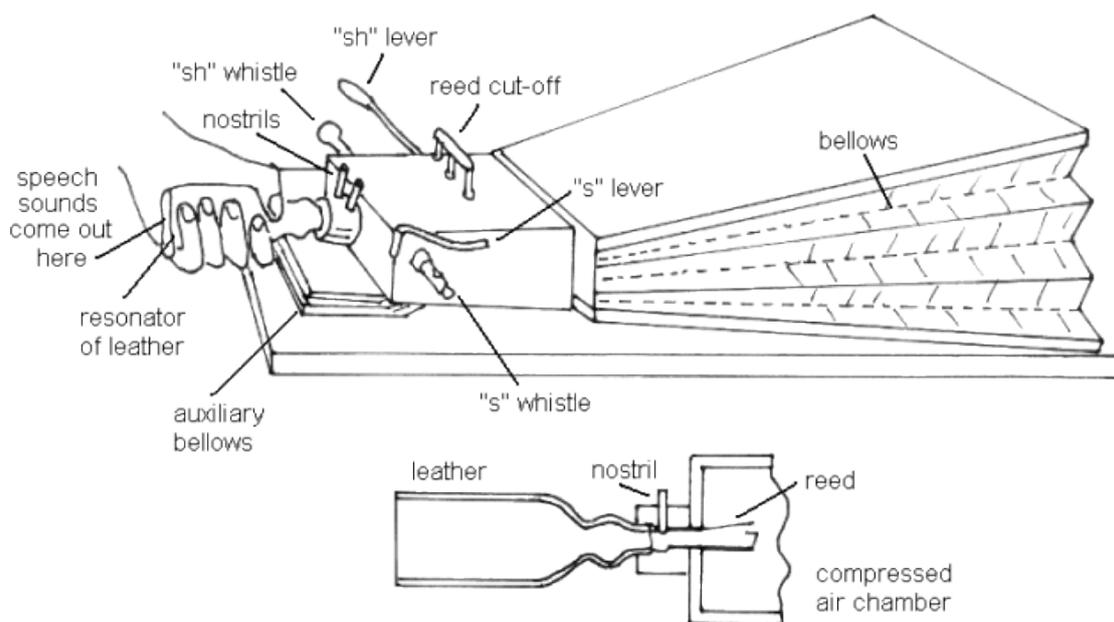


Figura 7 – Sintetizador de Voz com Fole de Kempelen.

Ao longo do tempo, o interesse em sintetizadores de voz se manteve. Inicialmente construídos como sistemas mecânicos, foram evoluindo até que pudessem analisar texto, passando a usar sistemas eletrônicos de codificação, como o ASCII, por exemplo.

Nas décadas de 1960 e 1970 essa ideia inicial foi desenvolvida: o MITalk podia converter texto em fala [HOL01] e a partir deste momento a ideia de que um sintetizador de voz era simplesmente falar palavra por palavra, convertê-las uma por uma em texto e concatená-las em uma *String* final deu lugar a real complexidade do problema [SO99]. A pronúncia do leitor, a ênfase em determinadas palavras e a não ênfase em outras, a variação de entonação entre os diferentes leitores criou novas possibilidades de pesquisa e desafios.

3. METODOLOGIA DE PESQUISA

“Pensar é o trabalho mais difícil que existe. Talvez por isso tão poucos se dediquem a ele.”

Henry Ford

O desenvolvimento do trabalho está dividido em duas grandes partes principais, chamadas de Fase 1 e Fase 2, que serão apresentadas a seguir em uma visão geral. Nos capítulos que seguem, estas fases serão explicadas de forma detalhada bem como os experimentos realizados em cada uma delas. A Figura 8 a seguir apresenta o desenho de pesquisa.

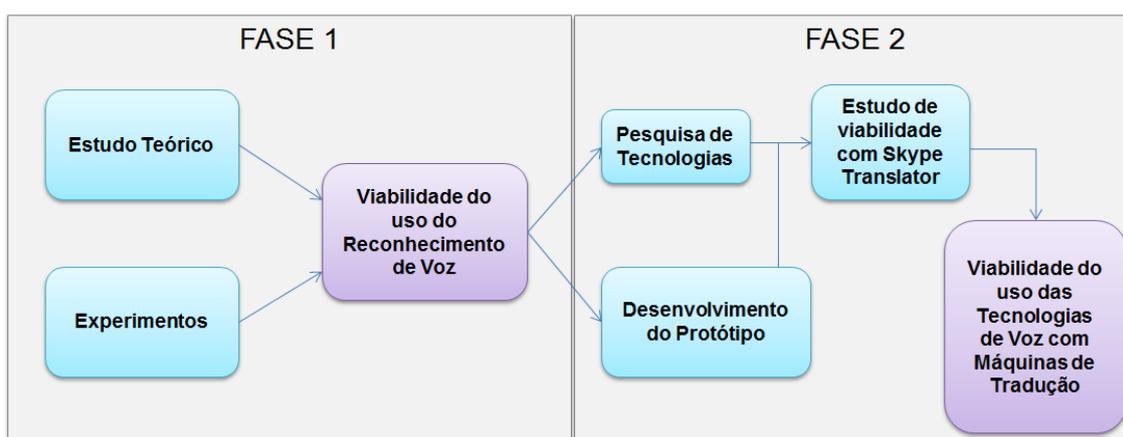


Figura 8 – Desenho de pesquisa.

A Fase 1 é composta por outras duas partes: uma que envolve a continuação do estudo teórico iniciado por [DUA14], onde foi realizada uma investigação sobre a existência de tecnologias de tradução automática de voz para o português brasileiro e posterior investigação exploratória (pesquisa *ad hoc*) na internet cujo objetivo era encontrar tecnologias comerciais ou que não tivessem sido exploradas no ambiente acadêmico. Depois de feita essa verificação inicial, foram encontradas ferramentas capazes de cumprir o objetivo e a realização da segunda parte foi possível, sendo composta por dois experimentos.

No primeiro experimento (veja Apêndice - Protocolo de Estudo 1) buscou-se compreender como erros no processo de reconhecimento de voz afetavam as saídas na máquina de tradução. Testes foram feitos usando aparelhos *mobile* e o *Google Translate Service*. O resultado deste estudo foi publicado em 2014 [CAL14b]. O segundo experimento (veja Apêndice - Protocolo de Estudo 2) uniu tecnologias de reconhecimento de voz e tradução do Google, respectivamente o *Google Web Speech API* e o *Google Translate Service*. Este foi um estudo empírico, onde 8 participantes usaram uma lista com 60 sentenças reais de requisitos de software. Este segundo experimento foi publicado em [CAL14].

Estes resultados da investigação teórica e dos estudos empíricos serviram como base para a Fase 2. Nesta fase uma nova pesquisa sobre a evolução das tecnologias foi feita, para verificar se algum serviço ou nova *feature* teria sido integrada às ferramentas no intervalo de tempo entre o trabalho de Duarte [DUA14] e o atual. Além disto, o desenvolvimento de dois protótipos que uniram o reconhecimento de voz, a tradução para um determinado idioma destino e a síntese desta tradução para voz foram feitos e analisados. Ao final desta análise, tendo em vista a dificuldade com as tecnologias utilizadas nos protótipos e considerando o lançamento do Skype Translator, desenvolvemos um estudo de viabilidade com esta tecnologia, de forma alternativa aos protótipos, visto que o Skype Translator oferecia integração entre os três componentes existentes em um processo de tradução em tempo real [SKY14a].

4. RESULTADOS DOS EXPERIMENTOS E DOS ESTUDOS

“Vá até onde sua vista alcançar e, ao chegar lá, você conseguirá enxergar mais longe.”

J. P. Morgan

Ao longo deste trabalho dois experimentos, dois protótipos e uma avaliação sobre o uso do *Skype Translator* foram desenvolvidos e, conforme mencionado no primeiro capítulo, a pesquisa proposta deu continuação aos trabalhos de Calefato [CAL10] e Duarte [DUA14] [DUA14b]. Os experimentos que serão detalhados a seguir foram planejados e desenvolvidos em parceria com os pesquisadores Fabio Calefato e Filippo Lanubile, da Universidade Aldo Moro de Bari, no contexto de um projeto apoiado pela FAPERGS e pelo CNPq. O projeto destes experimentos foi elaborado pelos pesquisadores italianos e a execução foi realizada nas Universidades Aldo Moro e na PUCRS.

Já o projeto dos protótipos e desenvolvimento dos mesmos foram elaborados na PUCRS. Os protótipos serviram para avaliar de forma prática as ferramentas, bibliotecas e o estado atual das tecnologias disponíveis, bem como o quão longe uma integração entre elas era possível. Os resultados obtidos são apresentados logo após o detalhamento dos experimentos.

Ao final deste capítulo, são apresentados também três diagramas com as principais ferramentas disponíveis no mercado. Explorou-se as características fundamentais destas ferramentas além de se identificar as dificuldades de uso e integração delas nos protótipos, ao mesmo tempo em que os experimentos eram executados.

4.1. EXPERIMENTO 1 – MOBILE

Neste primeiro experimento (veja Apêndice - Protocolo de Estudo 1) buscou-se compreender como erros no processo de reconhecimento de voz

afetavam as saídas na máquina de tradução. Os testes foram realizados com o uso de aparelhos *mobile*, sendo utilizado um Samsung Galaxy Note 3 rodando Android 4.3 no lado italiano e um Samsung Galaxy Note GT N7000 rodando Android 4.1.2 no lado brasileiro. Um conjunto de 51 sentenças selecionadas de *logs* de conversas em inglês foram coletadas pelos pesquisadores italianos em cinco diferentes *workshops* de requisitos de SW para serem usadas nestes testes [CAL12].

Para prosseguir com a investigação, foram escolhidos 3 idiomas como base para o estudo: o português, o italiano e o inglês. Sendo assim, 6 pares de idiomas foram criados (tabela 1) e a notação na tabela é interpretada da seguinte forma: EN(PT) >>> IT significa que determinada frase está sendo lida em inglês por um nativo em português e o idioma final será o italiano, já em IT(IT) >>> EM a frase está sendo lida em italiano por um nativo em italiano e o idioma final será o inglês.

Combinação dos pares de idiomas.
1. IT _(IT) >>> EN
2. PT _(PT) >>> EN
3. EN _(IT) >>> PT
4. EN _(PT) >>> IT
5. IT _(IT) >>> PT
6. PT _(PT) >>> IT

Tabela 1 – Combinação dos Pares de Idiomas.

O universo de testes é composto então por essas 51 sentenças que são de tamanhos variáveis de no mínimo 4 palavras até um máximo de 94 palavras, conforme o exemplo apresentado na tabela 2 abaixo.

Sentença	Número de palavras da sentença
Many patients check in but not all stay for a long period of time.	14
How will web users be authorized to access patient information on the hospital website? Or will there be no information they can access.	23
Welcome to the first elicitation meeting. We are going to get started here shortly, but in the meantime feel free to state your name and role. Is there anyone missing that requires an invite?	34
I was still trying to focus on these terminals, which are located inside of the hospital. That said, it would be my understanding that there is no need for a map of the location of the hospital on the terminals, but perhaps floor plans would be a better idea.	49
Welcome to the first elicitation meeting. The goal of this first meeting between St. Peter's Hospital and Tri-Systems is to bring both parties to collaborate and aid Tri-Systems understand the requirements of system outlined in the RFP. Jane and I are software developers for Tri-Systems. I am John and I will be moderating this meeting. Odysseus will be recording all decisions in the Decision place at the right side of your screen. If at any time you do not agree with one of the decisions please send a message and we will discuss further.	94

Tabela 2 – Alguns Exemplos de Sentenças Utilizadas no Experimento 1.

Além disso, o processo de avaliação criado para o reconhecimento de voz levou em conta a correta identificação das palavras enquanto estas eram faladas, ou seja, o quão preciso o resultado final foi em comparação com a frase original; já o processo de avaliação da máquina de tradução levou em conta a fidelidade da tradução em si. Para o cálculo da precisão a seguinte fórmula foi criada:

$$P = \frac{\#palavrasReconhecidas - \#palavrasErradas}{\#palavrasNaFrase}$$

Sendo $P \in [-1, 1]$, para apresentar os valores em porcentagens a normalização é feita como $P' = (P + 1) / 2$. Dessa forma, considerou-se que uma transcrição precisa teria uma taxa de acertos entre 75% e 100% e uma

transcrição não precisa ter uma taxa de acertos entre 0% e 75%, conforme apresentado na tabela 3.

Após a criação destes parâmetros, passamos para a fase dos testes propriamente ditos, onde um usuário utilizava então o aplicativo *Google Translate* de seu aparelho *mobile* para fazer o reconhecimento das sentenças. As sentenças eram lidas de um documento e ditadas pelo voluntário para o dispositivo. Este fazia o reconhecimento e a transcrição das frases, que eram traduzidas pelo aplicativo. Na figura 9 abaixo pode-se verificar como foi feito o processo.



Figura 9 – Uso do *Google Translate* no Dispositivo *Mobile*.

Ao final esta etapa, os resultados eram passados um a um para uma planilha, onde foram classificados conforme as explicações contidas na tabela 3.

Valor	Descrição
4	Completamente Preciso: A transcrição/tradução reflete claramente a informação contida na frase original. A frase resultante é perfeitamente entendível, clara, gramaticalmente correta e a leitura é de fácil entendimento.
3	Bastante Preciso: A transcrição/tradução reflete a maior parte da informação contida na frase original. Alguns erros são encontrados no texto. A frase é geralmente clara e inteligível e o leitor pode quase que imediatamente entender o significado.
2	Pouco Preciso: A transcrição/tradução reflete pobremente a informação contida na frase original. A sentença contém erros gramaticais e escolhas erradas das palavras. A ideia geral da frase é entendida depois de um estudo considerável.
1	Completamente Impreciso: A transcrição/tradução não reflete a informação contida na frase original. Tentar entender o sentido da sentença é inútil, a informação original é completamente perdida no processo.

Tabela 3 – Precisão do Reconhecimento de Voz e da Tradução.

Os seguintes resultados foram obtidos, onde cada coluna do gráfico representa um total de 51 sentenças analisadas pelo reconhecedor de voz.

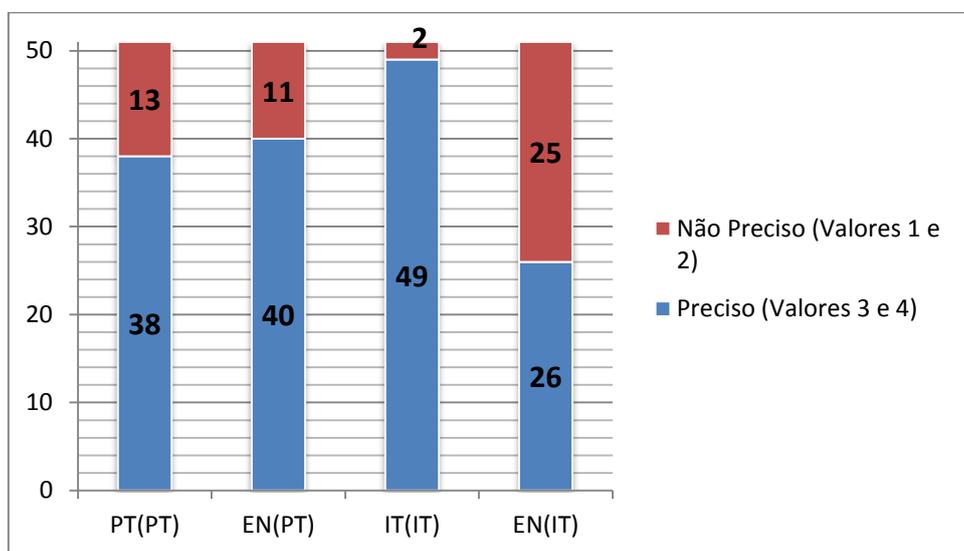


Gráfico 1 – Precisão do Reconhecimento de Voz.

Apesar do resultado ter sido considerado muito bom, no momento que a tradução foi feita, não obtivemos a mesma taxa de qualidade, onde no reconhecimento de voz, uma vez que consolidados os resultados, de um total de

204 sentenças, 153 pontuaram entre 3 e 4, chegando a uma taxa de 75% de precisão.

Na tradução os resultados foram os seguintes:

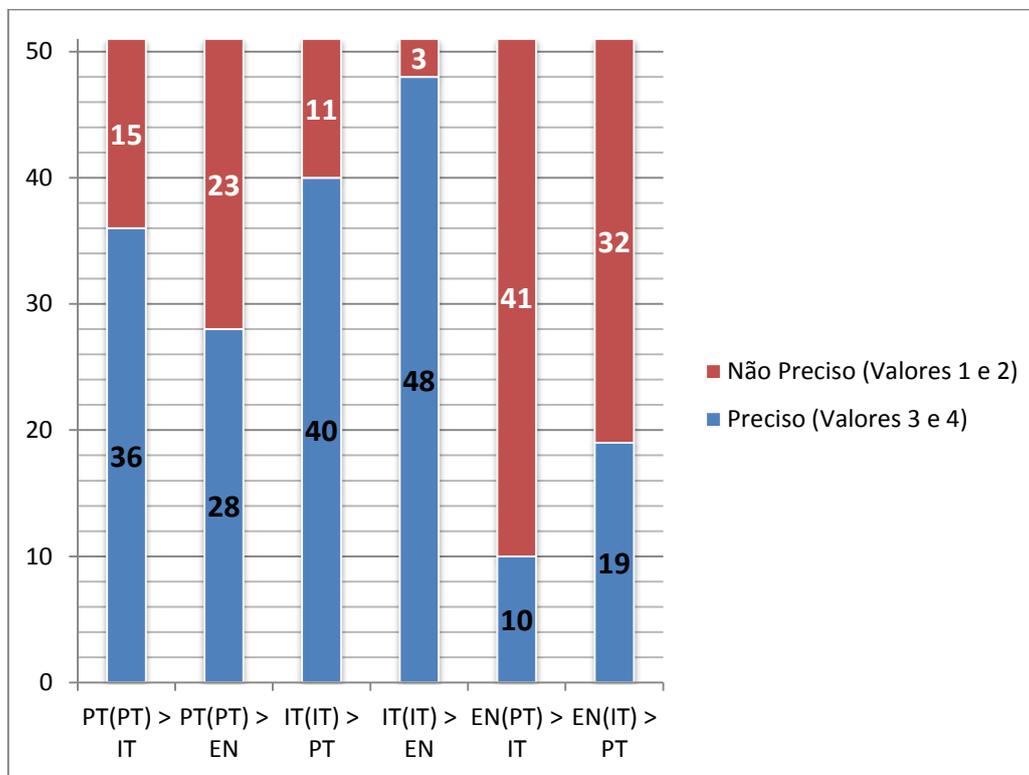


Gráfico 2 – Precisão das Traduções.

Dessa forma, das 306 sentenças que serviram como entrada para o processo de tradução os resultados que foram obtidos são de 181 com pontuação entre 3 e 4 e 125 com pontuação entre 1 e 2, o que resulta em 59% de precisão, outro valor considerado bom, mas bem abaixo do obtido no reconhecimento de voz.

Com estes valores, entendemos que seria importante testar em outra tecnologia que não fosse a disponível pelos aparelhos *mobile* e também continuar a buscar tecnologias e resultados que provassem que o reconhecimento de voz em conjunto com a tradução pode ser usado para ultrapassar as barreiras do idioma entre equipes distribuídas. Os resultados do Experimento 1 também resultaram em [CAL14b].

4.1.1. ANÁLISE CRÍTICA

Partindo-se do princípio que este primeiro experimento serviu para compreender como os erros no processo de reconhecimento de voz afetavam as saídas na máquina de tradução, ao olharmos para os resultados podemos notar que, de fato, quanto pior o reconhecimento de voz pior será o resultado da tradução nas máquinas de tradução. Além disso, esta lógica pode ser esperada tanto em máquinas de tradução que tenham como método a tradução por regras como também as máquinas que utilizam estatística para determinar as saídas para um determinado idioma. Soma-se à isto o fato de que os testes foram feitos em um grupo multilíngue, onde alguns dos participantes podem ter limitadas habilidades de comunicação e compreensão do inglês. Entretanto, acreditamos que este fator torna ainda mais real a comunicação em grupo que envolve pessoas que não são nativas no idioma inglês, fazendo com que o cenário possa representar de fato uma equipe distribuída de desenvolvimento de software.

Também foram analisadas algumas possíveis ameaças à validade deste estudo, pois, por se tratar de uma simulação, o universo dos testes, dos recursos e das tecnologias são limitados. Sendo assim, a primeiro fator que levamos em consideração foi o uso de apenas um sistema de tradução nos celulares, no caso o *Google Translate Mobile*. Isto poderia impactar diretamente os resultados. Dessa forma no segundo experimento expandimos este universo para a *web*. Levamos em consideração também o fato que neste experimento foram envolvidos apenas dois avaliadores, um italiano e um brasileiro. No entanto o conjunto de teste incluiu frases em italiano, português e inglês e nos resultados não obtivemos resultados negativos pelo fato de não termos um avaliador nativo em inglês. Por fim, os testes envolveram um conjunto de sentenças recolhidas de reuniões reais da área de engenharia de software, que foram realizadas em inglês. Embora os resultados sejam bons, não é possível concluir que estes resultados sejam suficientemente capazes de permitir aos participantes concluir com sucesso uma tarefa que envolva trabalho em grupo, como o desenvolvimento de um sistema de forma distribuída.

4.2. EXPERIMENTO 2 – WEB

O segundo estudo contou com a participação de oito voluntários. O objetivo neste caso era verificar se jargões ou termos comuns influenciavam no processo de reconhecimento e tradução de alguma forma (veja Apêndice – Protocolo de Estudo 2). Foi usado o *Google Web Speech API* para o reconhecimento de voz, uma tecnologia *web* disponível para o *Google Chrome*, conforme a figura 10 abaixo:

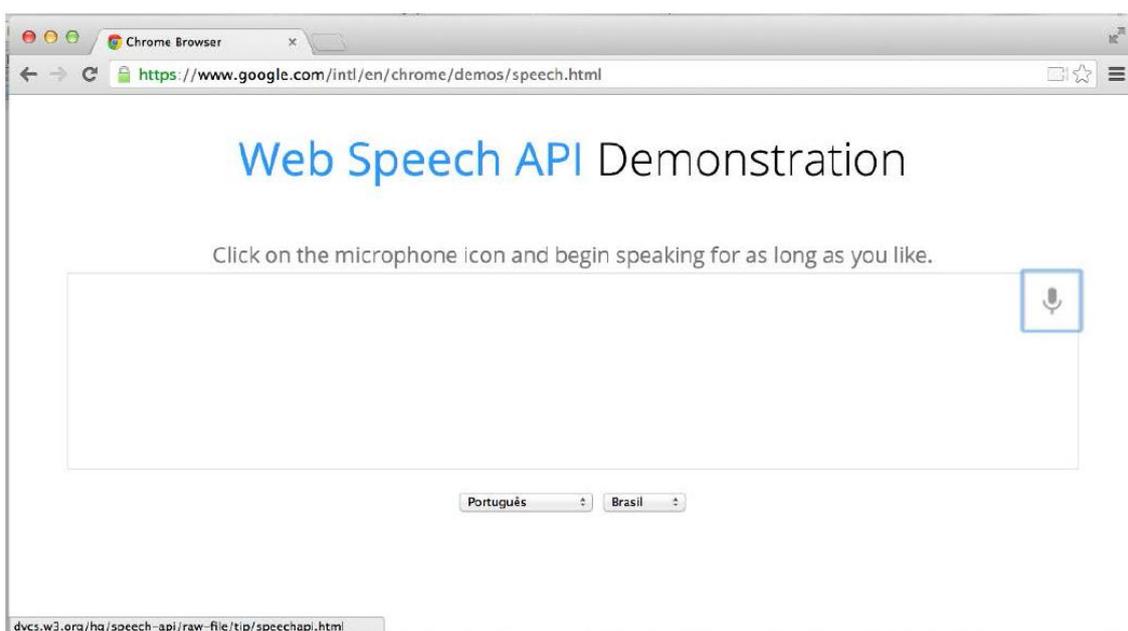


Figura 10 - Google Web Speech API.

Para a tradução foi usado o já conhecido *Google Translator* e neste experimento selecionamos mais 60 frases dos *workshops* de requisitos de SW, mas dessa vez elas continham de 5 até 30 palavras cada. Usando as mesmas combinações para a tradução que o estudo anterior, tem-se um número de 960 sentenças testadas no total e os resultados obtidos foram os seguintes:

- Precisão de 77.75% no reconhecimento de voz.
- Precisão de 35.62% na tradução.

Apesar do bom valor obtido no reconhecimento de voz, a tradução resultou num valor inesperado, porém isto não afetou o objetivo do trabalho, que

era verificar se jargões ou termos comuns eram identificados de formas diferentes. Quando se compara o valor obtido no reconhecimento de voz e tradução dos jargões, obtêm-se a precisão de 33.54% em comparação a 37.70% dos termos comuns. Estes valores nos levaram a duas perguntas:

1. A tradução de voz funciona com frases ininterruptas de forma aceitável?
2. Jargões técnicos afetam a tradução?

A primeira questão pode ser respondida levando em consideração o fato que nosso processo é feito utilizando a tradução de transcrições: podem ocorrer traduções inadequadas quando o reconhecimento é preciso, mas quando o reconhecimento de voz não é feito de forma precisa, a tradução nunca será efetiva, então o processo se torna fortemente atrelado a um bom reconhecimento de voz. Quanto à tradução de jargões técnicos, os resultados nos indicam que eles não têm tanta influência nas frases, ou seja, até mesmo quando as palavras são mantidas iguais (jargões técnicos que não foram traduzidos), o usuário acaba entendendo o significado das sentenças pelo seu contexto.

4.2.1. ANÁLISE CRÍTICA

Neste experimento, além de migrarmos para um sistema que não utilizasse aparelhos *mobile* para o reconhecimento de voz nem para a tradução, identificamos um caminho que não havíamos encontrado na literatura até então, que era o uso de jargões como um impeditivo para o bom entendimento do reconhecimento de voz e das traduções. Ao compararmos os dados deste experimento com o anterior, notamos que o reconhecimento de voz manteve uma alta taxa de acertos, porém a ideia inicial de que com um bom reconhecimento de voz teríamos também uma boa tradução não foi mantida. Apesar desta observação ter sido de importante, observamos que os participantes tinham sotaques diferentes do que no experimento 1 e este fator pode ter causado essa diferença nas comparações. Além disso, nossos resultados apontam que jargões técnicos não afetaram a tradução nas máquinas

de tradução, onde os voluntários mantiveram seu entendimento sobre aquilo que estávamos testando.

Outro fator importante que foi discutido inclui como comparar os resultados dos experimentos, já que pessoas tem sotaques diferentes, falam com velocidades que alternam de tempos em tempos e também tem timbres de voz únicos. Além disso, os testes realizados contam com as características disponíveis nos dispositivos dos voluntários e por isso a comparação entre resultados foi feita em termos de precisão e exatidão das palavras. Estes outros fatores que foram citados não estão sendo explorados neste trabalho, pois isto criaria um conjunto de possibilidades que não seria possível analisar e testar em tempo hábil.

4.3. PROTÓTIPO 1 – WEB

Na Fase 2 da metodologia iniciou-se o desenvolvimento de dois protótipos. O primeiro deles foi um sistema que integrava Reconhecimento de Voz, Máquina de Tradução e Síntese de Voz. A ideia inicial era que o sistema fosse um *web chat*, onde, através do *browser*, um usuário pudesse conversar com outro usuário, escolhendo os idiomas origem e destino, pois até o momento todos os experimentos que haviam sido realizados foram manuais, onde a saída do reconhecimento de voz era informada para o tradutor por um usuário e assim por diante.

Dessa forma, com base em todo conhecimento que tínhamos adquirido até aquele momento e com os resultados das pesquisas de Duarte [DUA14] [DUA14b], fizemos a escolha das seguintes tecnologias para este protótipo *Web*:

- Google Web Speech API – como reconhecedor de voz;
- Google Translate API – como máquina de tradução;
- W3C Web Speech API – como sintetizador de voz;
- JSP – para a criação das páginas do protótipo;
- Javascript – para integração entre os componentes;
- Glassfish Server 4 – como servidor de aplicação.

Após o início do desenvolvimento, a integração entre o reconhecedor de voz escolhido e a máquina de tradução se deu de forma precisa, onde as frases reconhecidas eram passadas de forma automatizada para o tradutor através de *callbacks* das funções e este retornava a frase traduzida. Até aquele momento, estávamos utilizando o Google Chrome como *browser* para os nossos testes locais. Ao testarmos o protótipo em outros *browsers*, notamos que o Google Web Speech API é dependente do Google Chrome para funcionar, ou seja, não seria possível utilizar nenhum outro *browser* para abrir o protótipo, pois este não teria o componente do microfone funcionando com os componentes do Google Chrome. Na figura 11 abaixo podemos notar o pedido de permissão para acessar o microfone que é feito pelo *browser* do Google.

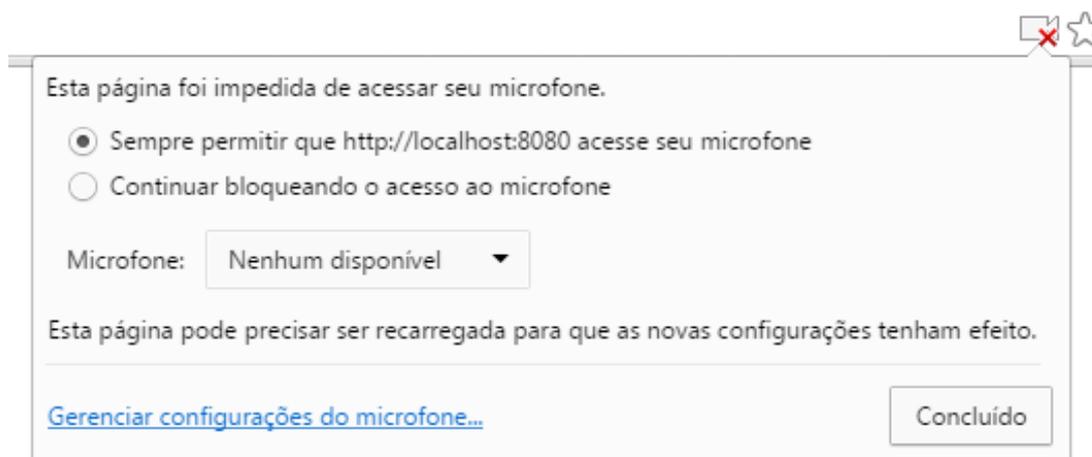


Figura 11. Permissão para acesso ao microfone.

Apesar deste detalhe, seguimos para a integração desta parte com o sintetizador de voz, que ocorreu de forma simples, onde a saída da máquina de tradução era passada por parâmetro para o sintetizador e bastava “setar” os outros parâmetros de idioma destino, velocidade da fala e volume, conforme a função criada abaixo mostrando no detalhe em inglês:

```
$("#ReproduzirEmIngles").on('click', function(e) {  
    var texto = document.getElementById('translation').innerHTML;  
    var sintetizador = new SpeechSynthesisUtterance(texto);  
    sintetizador.lang = 'en-US';
```

```
sintetizador.rate = 1;
sintetizador.volume = 1;
speechSynthesis.speak(sintetizador);
});
```

Com isto, já tínhamos os 3 componentes integrados, como podemos ver na figura 12, faltando a criação do *chat*, porém alguns problemas maiores que a questão do *browser* começaram a aparecer.



Figura 12. *Print Screen* do protótipo *web*.

O primeiro problema é que o sintetizador da W3C ainda não tem suporte ao idioma português, que é essencial para o trabalho, já que a parceria é entre Brasil e Itália. Outra questão importante é que as bibliotecas de comunicação em Java para o *chat* não ofereciam suporte *web* e por fim, o tempo que o sintetizador levava para interpretar e “falar” era em média de 3 segundos por frase, o que tornou o retorno lento e de baixa produtividade em testes feitos localmente. Estes

fatores nos levaram a pausar o desenvolvimento *web* e partir para uma nova ideia, já que os componentes do *chat* funcionam para *desktop*.

Embora o desenvolvimento deste protótipo não tenha resultado na criação de uma ferramenta totalmente funcional, ele serviu para nos mostrar as dificuldades que teríamos e também para mapear quais bibliotecas e tecnologias disponíveis podem se integrar e quais incompatibilidades existem.

4.4. PROTÓTIPO 2 – JAVA DESKTOP

Os fatores apresentados na sessão anterior fizeram com que migrássemos para uma estratégia *desktop* e para isto os pesquisadores italianos sugeriram o uso do *Skype4Java*, uma biblioteca escrita em Java que serve de ponte para os principais métodos e para as camadas de rede do *Skype*, facilitando então o desenvolvimento de um *chat*. Neste protótipo, a estratégia de começar pela integração deu lugar para iniciarmos com o *chat*, onde essa biblioteca realmente facilitou e aumentou a produtividade para o desenvolvimento. As tecnologias utilizadas são as que seguem:

- *Skype4Java* – para a conexão com usuários e servidor;
- Java 7 – como linguagem de programação para o protótipo;
- Google Web Speech API – como reconhecedor de voz;
- Google Translate API – para a máquina de tradução.

Com o protótipo do *chat* desenvolvido, começamos com alguns testes locais, para verificar o andamento da ferramenta, verificar se os usuários do *Skype* eram carregados e se as trocas de mensagens eram efetivas. Desta vez tudo ocorreu da forma esperada, conforme exemplos nas figuras 13 e 14 abaixo.

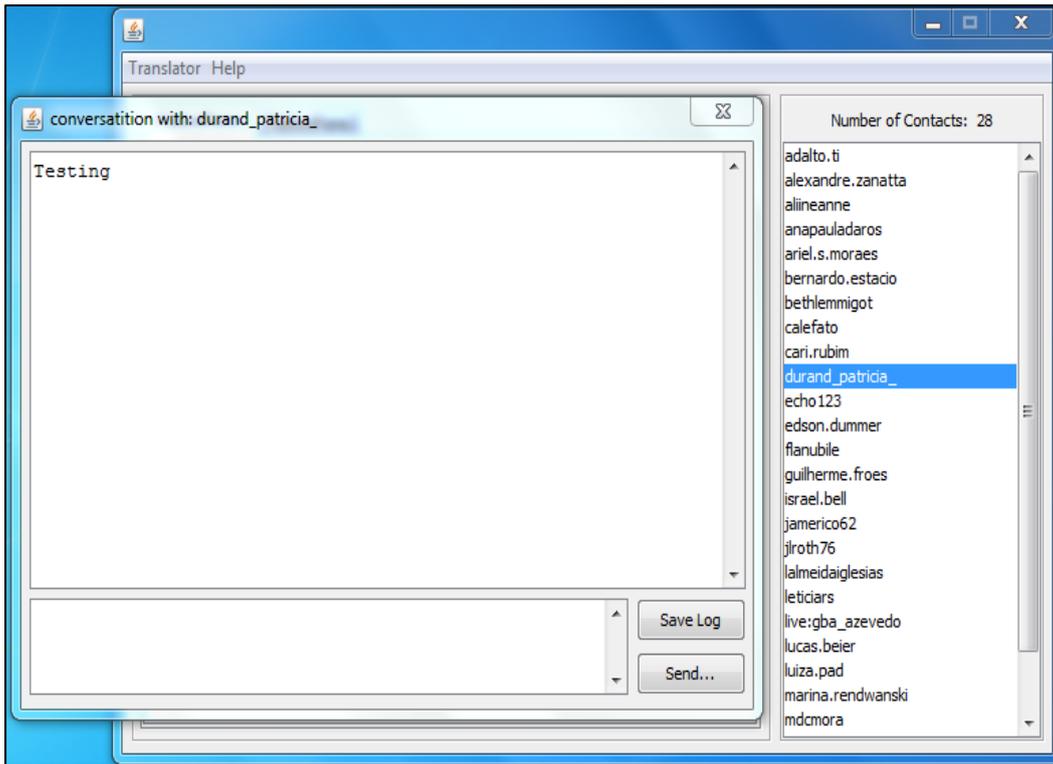


Figura 13. Lista de amigos do *Skype* funcionando no protótipo.

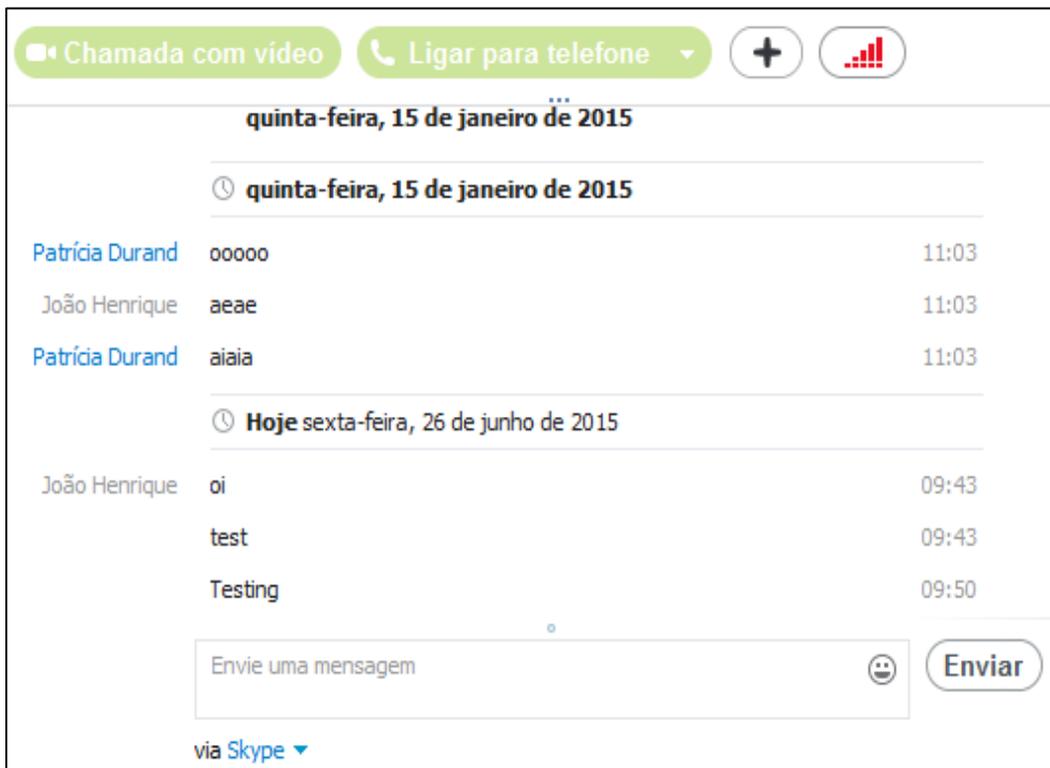


Figura 14. Mensagem sendo recebido pelo próprio cliente do *Skype*.

Com o bom funcionamento do *chat*, começamos a integrar os outros três componentes no protótipo, iniciando-se pela máquina de tradução. A biblioteca escolhida para isto foi a *google-api-translate-java-0.97.jar*, e o resultado da máquina de tradução integrada com o *chat* cumpriu este objetivo, onde dois ou mais usuários podiam se comunicar de forma textual através das mensagens que eram traduzidas em tempo real, com a ajuda do versão 2 desta *API* do Google. O usuário escolhia o seu idioma origem e o idioma destino e a tradução era feita de forma instantânea ao chegar no destino.

Ao colocarmos mais o reconhecimento de voz, nos deparamos com o problema do *browser*, sendo assim uma tentativa de renderizar o browser via Java foi feita com o uso do *JEditorPane*, um componente específico para este tipo de função, porém sem sucesso. O mesmo problema da necessidade de rodar o Reconhecedor de Voz pelo Google Chrome aconteceu, sendo assim resolvemos por encerrar as tentativas de criar um protótipo e manter os testes com ferramentas já existentes, pois todas as tentativas de integração chegaram até certo ponto, mas não foi possível realizar a integração completa dos 3 componentes mais o *chat*.

4.5. ESTUDO DE VIABILIDADE TÉCNICA – SKYPE TRANSLATOR

Um estudo de viabilidade usando o *Skype Translator* ocorreu entre voluntários italianos e brasileiros, porque naquele momento esta ferramenta surgia como uma real opção [NYT15] [DAI15] para a continuação do desenvolvimento dos protótipos (em testes locais estes já davam indícios que não dariam suporte às necessidades desejadas) e, após testes preliminares entre as equipes, que serviram para avaliar a disponibilidade dos serviços, a qualidade da conexão e os tempos de resposta entre as mensagens, chegou-se à conclusão que a qualidade da conexão oferecida pelo *Skype Translator* ainda impossibilita testes em um ambiente real de desenvolvimento de software.

A ideia deste estudo era justamente testar de forma integrada todos os componentes que possibilitariam uma completa tradução em tempo real de uma conversa entre equipes distribuídas, completando todas as etapas necessárias para uma máquina de tradução voz, segundo [DUA14] [DUA14b] [WAI08] e conforme apresentado na figura 15 abaixo.

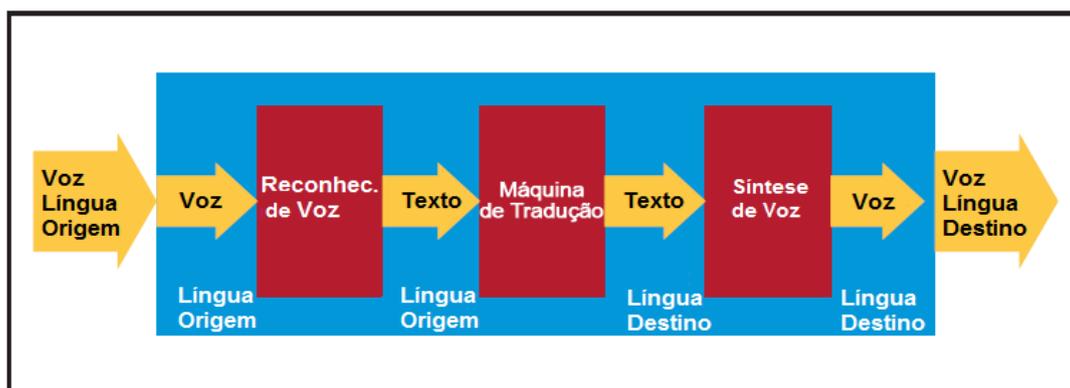


Figura 15 – Etapas de uma Máquina de Tradução de Voz.

Após diversas tentativas para se obter uma conexão estável entre um usuário italiano e um usuário brasileiro, acabamos por cancelar o mesmo e, embora as conexões com o reconhecedor de voz e com o sintetizador de voz não tenham sido possíveis, obtivemos um excelente resultado ao utilizarmos o *chat* com a máquina de tradução.

Estes testes preliminares acabaram por indicar que para os pares de idiomas IT-EN e EN-IT, é possível manter uma conversa entre pessoas que não entendem estes idiomas, seja esta uma conversa com termos técnicos de requisitos de *software* ou não. Foram feitas duas conversas, ambas com o professor Fabio Calefato e os *logs* das conversas foram analisados e discutidos entre a equipe. Na primeira conversa foram trocadas 100 frases, contendo um total de 578 palavras e foi usada a mesma tabela 3 de precisão para a análise. Abaixo os resultados obtidos nesse primeiro teste.

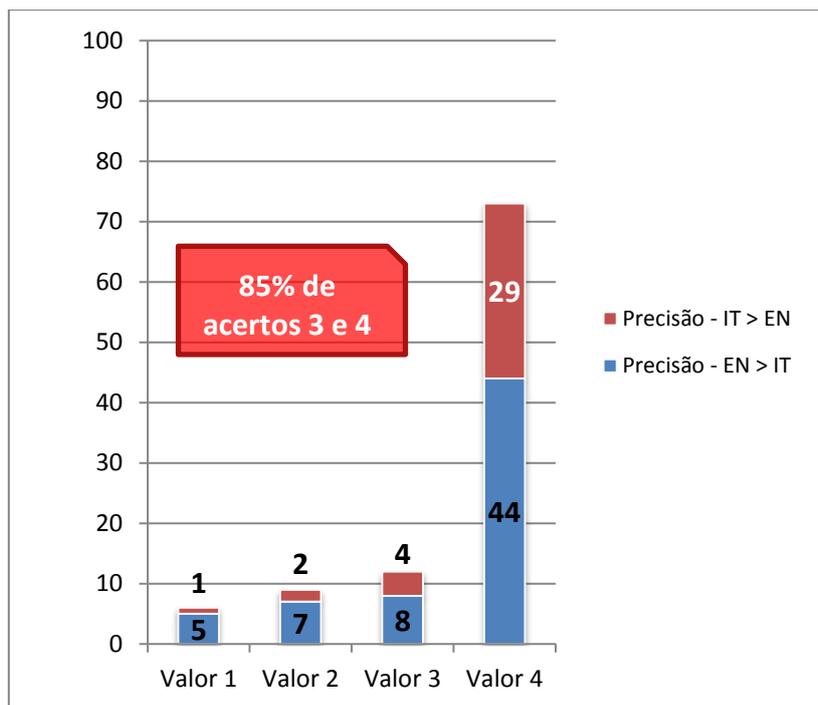


Gráfico 3 – Taxa de Acertos no teste preliminar 1 para IT-EN e EN-IT.

Após feita esta verificação, outras tentativas de conexão usando todos os componentes do *Skype* foram retomadas, porém os erros de conexão entre o reconhecedor de voz e a síntese de voz persistiram, dessa forma outra conversa pelo *chat* foi feita, com a finalidade de continuar a análise da ferramenta e, mais uma vez, os resultados foram bastante expressivos, conforme podemos notar no gráfico 4 abaixo:

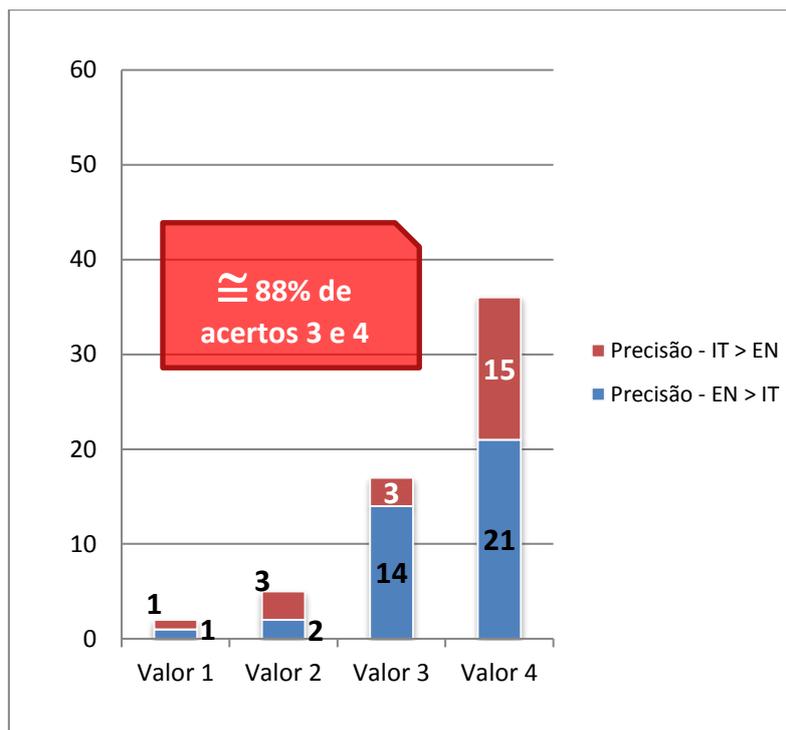


Gráfico 4 – Taxa de Acertos no teste preliminar 2 para IT-EN e EN-IT.

4.6. MAPA DAS TECNOLOGIAS PESQUISADAS

A seguir são apresentados diagramas contendo todas as tecnologias pesquisadas ou exploradas ao longo deste trabalho. Também foram levados em consideração os dados técnicos e parte da revisão sistemática do trabalho de Duarte [DUA14] para a identificação das melhores tecnologias atualmente disponíveis, bem como quais ferramentas poderiam ser descartadas por falta de recurso ou por inviabilidade técnica das mesmas. As tecnologias serão apresentadas em três grupos: Reconhecimento de Voz, Máquinas de Tradução e Síntese de Voz.

Nestes diagramas são apresentadas as principais características que cada ferramenta possui, bem como se a comunidade pode contribuir para a evolução de determinada ferramenta, quais os idiomas são suportados por qual ferramenta, se o projeto continua ativo ou se foi descontinuado e outras

características que podem levar desenvolvedores a descobrirem novos caminhos para a área de máquinas de tradução em tempo real.

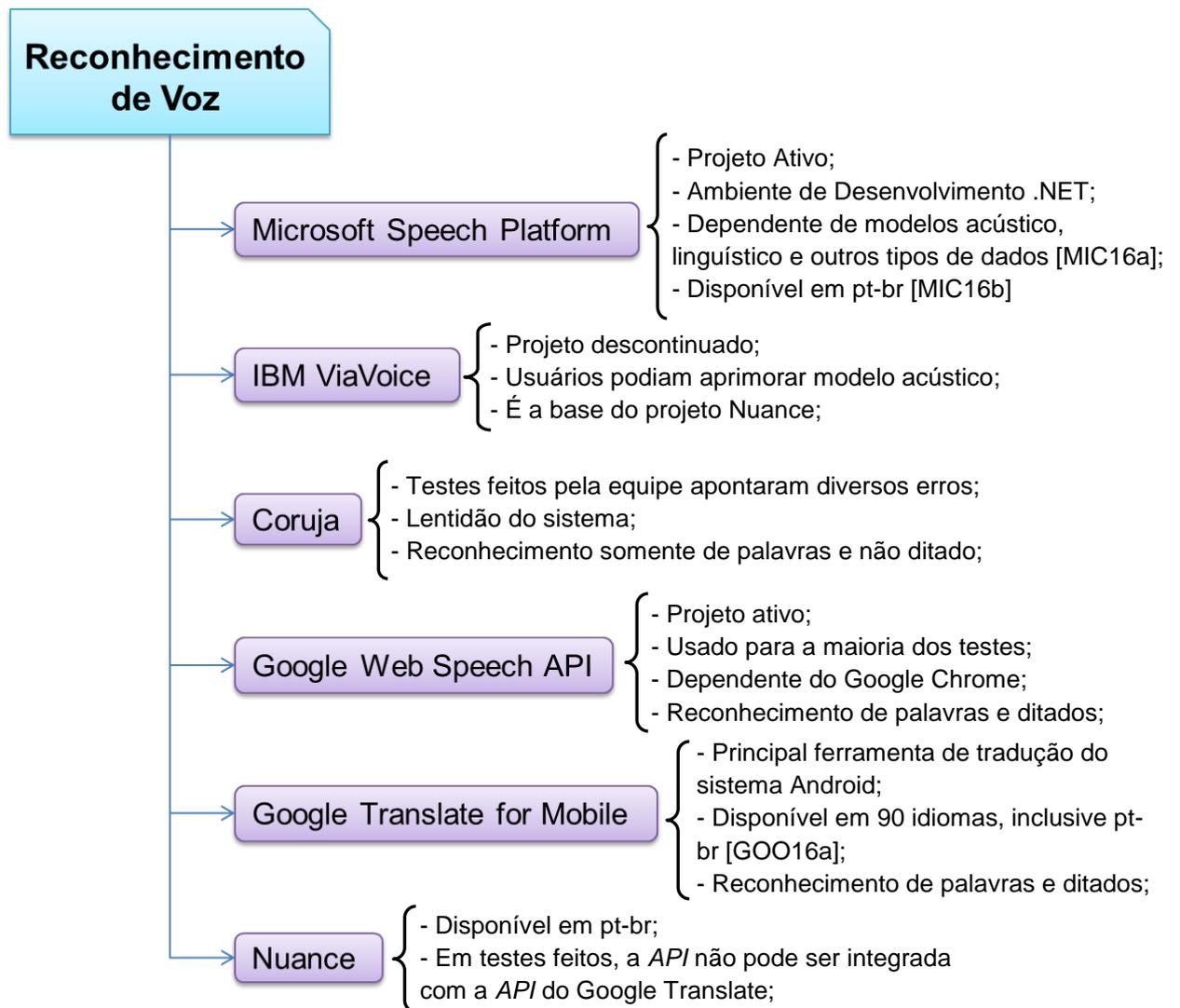


Figura 16 – Aplicações e Restrições das Tecnologias de Reconhecimento de Voz.

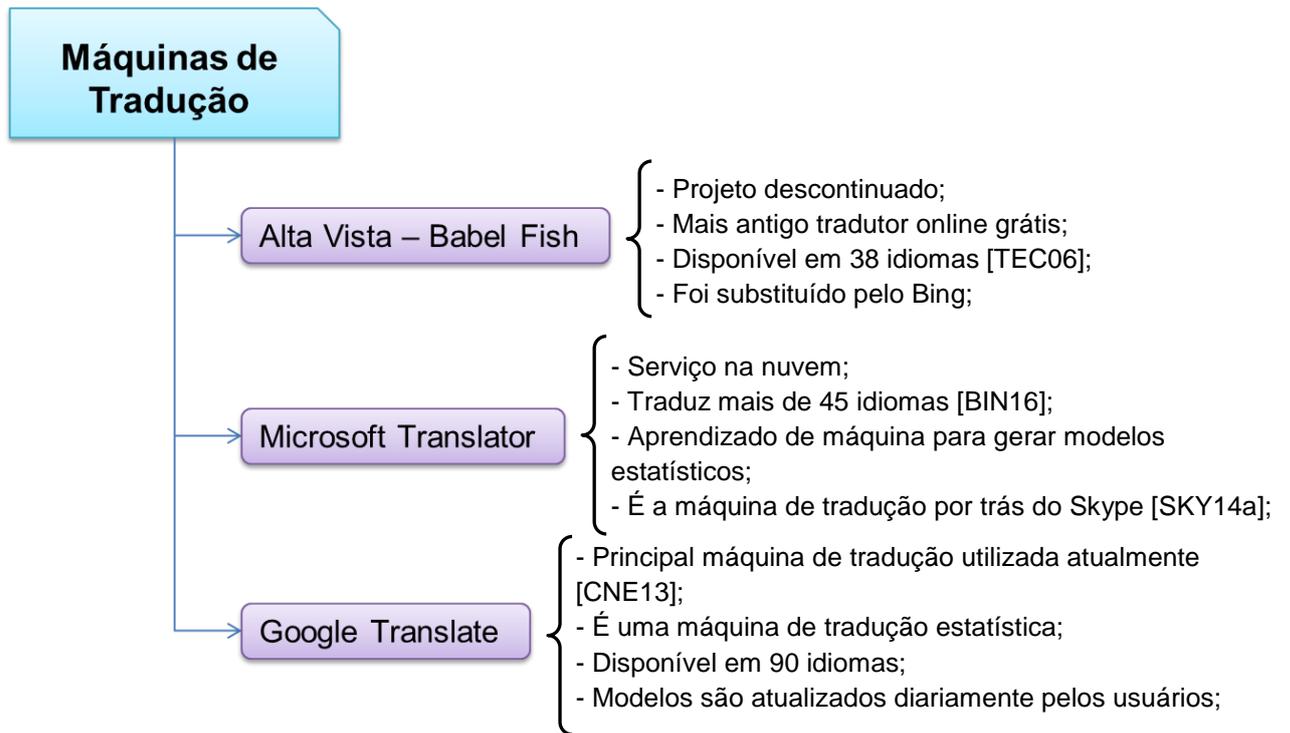


Figura 17 – Aplicações e Restrições das Máquinas de Tradução.

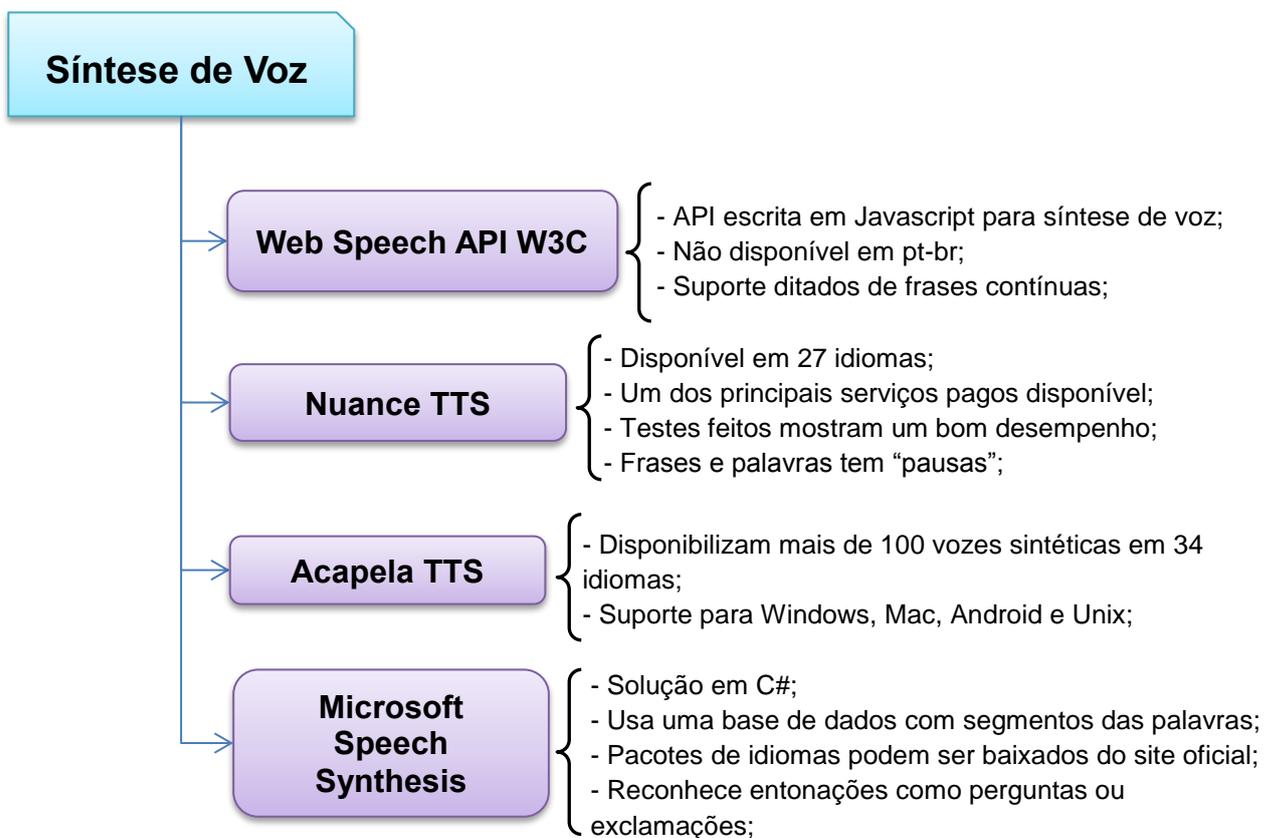


Figura 18 – Aplicações e Restrições dos Sintetizadores.

Nos diagramas apresentados, não incluímos o Skype nem a nova ferramenta *Skype Translator*, pois a primeira tem como função principal a comunicação em *chat*, já a segunda não foi incluída porque ela estaria classificada em uma categoria superior, dispondo de todos os três tipos de serviços que as demais tecnologias apresentam. Sendo assim nosso mapeamento de tecnologias passa por 14 diferentes serviços ou aplicações, além dos que foram descartados pelos motivos mencionados mais acima.

5. CONSIDERAÇÕES FINAIS

A globalização é uma característica fundamental atualmente, onde o software é um componente essencial [CAR01]. Cada vez mais as empresas estão distribuindo seus processos, diversificando seus ambientes e buscando formas de aumentar a qualidade e a produção e diminuir seus custos.

Máquinas de tradução em tempo real podem contribuir ainda mais para isto, pois além de disporem de serviços de vídeo ou áudio, geram *logs* que podem ser usados como documentação e também facilitam muito as já remotas reuniões e encontros entre clientes e provedores de serviço distribuídos fisicamente. Os resultados encontrados ao longo do estudo indicam que a área ainda pode crescer muito em termos de pesquisa e experimentos, pois as ferramentas continuam em constante evolução e, além disso, ao analisarmos o Hype Cycle do Gartner [GAR15], podemos comprovar esta tendência, possibilitando aprofundar os trabalhos sobre máquinas de tradução em tempo real no contexto do desenvolvimento distribuído de software.

5.1. CONTRIBUIÇÕES DA PESQUISA

A principal contribuição desta pesquisa foi conseguir fazer uma completa análise sobre o uso de máquinas de tradução em tempo real para equipes distribuídas de desenvolvimento de software no contexto da Engenharia de Software. Todos os testes e experimentos foram realizados com o uso de requisitos reais de reuniões com a finalidade de verificar a aplicação destas tecnologias no ambiente de desenvolvimento, práticas de gerência de software, em níveis de produtividade, comunicação e qualidade.

Junto a isto, outra contribuição desta pesquisa foi conseguir identificar um conjunto de ferramentas que podem vir a ser utilizadas para a comunicação entre equipes distribuídas que não falam o mesmo idioma. Entretanto, estas ferramentas ainda não disponibilizam todos os recursos suficientes – sejam eles idiomas, modelos acústicos ou conexão entre usuários, por exemplo. Sendo assim, esse conjunto some-se ao trabalho de Duarte [DUA14] e aos trabalhos

de Calefato [CAL10] [CAL11] [CAL12] [CAL12b] [CAL14] [CAL14b] como um modelo para ajudar a responder aos desafios da área de máquinas de tradução em tempo real no contexto do DDS.

A questão de pesquisa elaborada como forma de nortear o trabalho foi respondida ao longo do processo, onde pudemos verificar que o processo de tradução de voz em tempo real ainda se encontra em evolução e não seria indicado utilizá-lo em um projeto real de desenvolvimento de software, principalmente onde seja necessário o uso do idioma português.

Ao testarmos os protótipos e também as ferramentas, seja de forma individual ou de forma integrada, chegamos à conclusão que o entendimento entre as equipes pode resultar em um sistema ou módulos de um sistema se usada somente a máquina de tradução, pois ao usarmos o reconhecedor de voz e a síntese de voz, o número de fatores que levam ao sucesso aumenta de forma considerável e não se conseguiu viabilizar algo consistente dentro deste cenário.

É importante também destacar que esta pesquisa gerou três publicações em conferência, onde foram apresentados os resultados obtidos, bem como discussão de possíveis soluções, resultando em [CAL14] [CAL14b] [PIN15].

5.2. LIMITAÇÕES DA PESQUISA

Uma das principais limitações da pesquisa refere-se à falta do idioma português nas principais ferramentas de reconhecimento de voz e também de síntese de voz. Entretanto, podemos destacar que os resultados obtidos nos experimentos, quando se usou as tecnologias de forma individual nos idiomas inglês e italiano mostram um desempenho considerável em termos de qualidade e precisão das sentenças reconhecidas e sintetizadas. Se considerarmos que estes idiomas estão mais maduros, em algum momento o idioma português também estará disponível e poderá ser testado em conjunto com os demais idiomas, sendo assim temos um grau de segurança nas respostas das questões de pesquisa obtidas. Esta limitação também impactou no desenvolvimento dos protótipos,

onde a total integração entre o reconhecimento de voz, as máquinas de tradução e a síntese de voz não foi possível, podendo ainda ser explorado.

Outro fator que impactou em uma limitação para esta pesquisa é que existe uma grande quantidade de idiomas que não foram explorados devido à questões como falta de voluntários estrangeiros, custos de experimentos e dificuldades de se encontrar tempo comum entre os voluntários para testes. Estes fatores deixam em aberto uma série de idiomas que podem ser analisados em pesquisas futuras, verificando idiomas como o alemão, espanhol, francês e o hindi, por exemplo, devido à grande quantidade de profissionais que falam estes idiomas na área de Engenharia de Software.

Por fim, após pesquisas em bases da ACM Digital, IEEE e Scopus, foram encontrados pouquíssimos trabalhos relacionados, os quais podemos destacar mais uma vez as pesquisas feitas por Calefato [CAL10] [CAL11] [CAL12] [CAL12b] [CAL14] [CAL14b] e também o trabalho realizado por Duarte [DUA14].

6. PESQUISAS FUTURAS

Um dos objetivos principais da pesquisa era conseguir desenvolver um artefato de software com a ajuda de equipes distribuídas de desenvolvimento de software usando uma máquina de tradução em tempo real. Sendo assim, ao iniciarmos as conversas com os pesquisadores italianos, esse plano foi traçado e tentamos cumpri-lo ao longo desses últimos dois anos. Ao testarmos as ferramentas de forma individual, depois ao criarmos os protótipos *web* e *desktop* e por fim tentando por diversas vezes estabelecer conexões entre Brasil e Itália pelo Skype Translator, acabamos verificando que ainda não é possível desenvolver um projeto de forma distribuída usando uma máquina de tradução em tempo real.

Além disto, o fato do idioma português ser pouco explorado, quase nunca estar disponível nas mesmas datas de lançamentos das ferramentas normalmente usadas e também não ter a mesma qualidade de outros idiomas em termos de precisão e fidelidade das palavras, fez com que não pudéssemos utilizá-lo por alguns momentos, principalmente no reconhecimento e na síntese de voz. Sendo assim, um experimento utilizando uma nova versão do Skype Translator ou até mesmo de uma evolução das ferramentas do Google que disponibilizem a tradução do idioma português para o inglês ou espanhol, por exemplo, pode ser de grande utilidade para a área. Com isso, seria possível demonstrar que as equipes distribuídas conseguiriam desenvolver softwares mesmo que seus integrantes não conversem utilizando o mesmo idioma.

6.1. REFLEXÃO FINAL

Ao longo desta pesquisa diversos fatores se mostraram complexos na área de máquinas de tradução como um todo. Desde os primórdios da área, por volta do início do século 20 [LUC16] [RED76], a literatura já apresentava a dificuldade que tínhamos pela frente. Junto à isto, naquele tempo ainda não tínhamos a tecnologia necessária para criar modelos virtuais, sendo assim os

primeiros protótipos eram máquinas mecânicas que simulavam o comportamento humano para sintetizar a voz, por exemplo, ou ainda um tradutor mecânico que fazia o trabalho de uma pessoa.

Com a evolução e a globalização da nossa sociedade, hoje podemos levar um dispositivo de tradução no bolso e esse poder nos proporciona a oportunidade de explorar áreas como o desenvolvimento distribuído de software. O DDS, por sua vez, nos apresenta fatores como a diversidade cultural, e a dispersão temporal e geográfica, adicionando desafios ao desenvolvimento de software como era conhecido.

Juntando todos estes aspectos, acreditamos que apesar deste trabalho ter apresentado evidências das dificuldades de usar máquinas de tradução em tempo real para equipes de desenvolvimento de software, a tecnologia está evoluindo rapidamente. Neste sentido, entendemos que estamos perto de poder desenvolver projetos complexos de desenvolvimento de software com pessoas que não falam o mesmo idioma que o nosso, que estão em outro lado do planeta e apenas utilizando ferramentas automatizadas para esclarecer dúvidas de requisitos e interagir diariamente com as equipes.

7. REFERÊNCIAS BIBLIOGRÁFICAS

- [ALL90] Allen, J.; Hunnicutt, M. S.; Klatt, D. "From Text to Speech: The MITalk System", *Cambridge: Cambridge University Press*, 1987, pp. 261, *Lingua*, vol. 81-2, Jul 1990, pp. 265-270.
- [BIN16] Bing. "About Microsoft Translator". Capturado em: <https://www.bing.com/translator/help/#AboutMicrosoftTranslator>, Janeiro 2016.
- [CAL10] Calefato, F.; Lanubile, F.; Minervini, P. "Can Real-Time Machine Translation Overcome Language Barriers in Distributed Requirements Engineering?" In: *5th IEEE International Conference on Global Software Engineering*, 2010, pp. 257-264.
- [CAL11] Calefato, F.; Lanubile, F.; Prikladnicki, R. "A Controlled Experiment on the Effects of Machine Translation in Multilingual Requirements Meetings". In: *6th IEEE International Conference on Global Software Engineering*, 2011, pp. 94-102.
- [CAL12] Calefato, F.; Damian, D.; Lanubile, F. "Computer-Mediated Communication to Support Distributed Requirements Elicitations and Negotiations Tasks". *Empirical Software Engineering Journal*, vol. 17, no. 6, 2002, pp. 640-674.
- [CAL12b] Calefato, F.; Lanubile, F.; Conte, T.; Prikladnicki, R. "Assessing the Impact of Real-Time Machine Translation on Requirements Meetings: A Replicated Experiment". In: *6th ACM/IEEE International Symposium on Empirical Software Engineering and Measurement*, 2012, pp. 251-260.
- [CAL14] Calefato, F.; Lanubile, F.; Prikladnicki, R.; Pinto, J. H. S. "An Empirical Simulation-based Study of Real-Time Speech Translation for Multilingual Global Project Teams". In: *8th ACM/IEEE International Symposium on Empirical Software Engineering and Measurement*, 2014, pp. 56.
- [CAL14b] Calefato, F.; Lanubile, F.; Romita, F.; Prikladnicki, R.; Pinto, J. H. S. "Mobile Speech Translation for Multilingual Requirements Meetings: A Preliminary Study". In: *IEEE 9th International Conference on Global Software Engineering*, 2014, pp. 145-152.

- [CAR99] Carmel, E. "Global Software Teams – Collaborating Across Borders and Time Zones." Upper Saddle River, NJ: Prentice Hall, 1999, pp. 269.
- [CAR01] Carmel, E.; Agarwal, R. "Tactical Approaches for Alleviating Distance in Global Software Development". *IEEE Software*, vol. 18-2, 2001, pp. 22-29.
- [CAR02] Carmel, E.; Agarwal, R. "The Maturation of Offshore Sourcing of Information Technology Work". *MIS Quarterly Executive*, vol. 1, num. 2, 2002, pp. 65-77.
- [CAR05] Carmel, E.; Tija, P. "Offshoring Information Technology: sourcing and outsourcing to a global workforce". *Cambridge: Cambridge University Express*, 2005, pp. 96-100.
- [CM60] Corbé, M. "La Machine à Traduire Française Aura Bientôt Trente Ans". In: *Automatisme 5*, num 3, 1960, pp. 87-91.
- [CNE13] CNET. "Google Translate now serves 200 million people daily". Capturado em: <http://www.cnet.com/news/google-translate-now-serves-200-million-people-daily/>, Dezembro 2014.
- [DAI15] Daily Mail. "Skype Translator breaks barriers...". Capturado em: <http://www.dailymail.co.uk/sciencetech/article-2908490/Google-Translate-app-adds-conversation-mode-translate-SPEECH-real-time.html>, Fevereiro 2016.
- [DAM06] Damian, D.; Moitra, D. "Guest Editors' Introduction: Global Software Development: How Far Have We Come?". *IEEE Software*, vol. 23-5, 2006, pp. 17-19.
- [DD03] Damian, D.; Zowgui, D. "Requirements Engineering Challenges in Multi-Site Software Development Organizations". In: *Requirements Engineering Journal*, vol. 8-3, 2003, pp. 149-160.
- [DD07] Damian, D. "Stakeholders in Global Requirements Engineering: Lessons Learned from Practice". *IEEE Software*, vol. 24-2, Mar 2007, pp. 21-27.
- [DD11] Daw, D. "What Makes Siri Special?". *PCWorld*, Capturado em http://www.techhive.com/article/242479/what_makes_siri_special_.html, Dezembro 2014.

- [DRA14] Dragon Naturally Speaking. “History of Speech & Voice Recognition and Transcription Software”. Capturado em http://www.dragon-medical-transcription.com/history_speech_recognition_timeline.html, Dezembro 2014.
- [DUA14] Duarte, T. “Máquinas de Tradução Aplicadas à Comunicação em Tempo Real para Desenvolvimento Distribuído de Software”, Dissertação de Mestrado, Programa de Pós-Graduação em Ciência da Computação, PUCRS, 2014.
- [DUA14b] Duarte, T.; Prikladnicki, R.; Calefato, F.; Lanubile, F. “Speech Recognition for Voice-Based Machine Translation”. *IEEE Software*, vol. 31, Jan-Fev 2014, pp. 26-31.
- [EVA04] Evaristo, R.; Scudder, R.; DeSouza, K. “A Dimensional Analysis of Geographically Distributed Project Teams: A Case Study”. *Journal of Engineering and Technology Management*, vol. 21-3, 2004, pp. 75-189.
- [GAR15] Gartner Inc. “Gartner Hype Cycle”. Capturado em: <http://www.gartner.com/technology/research/methodologies/hype-cycle.jsp>, Janeiro 2016.
- [GOO15a] Google. “500 milhoes usuarios”. Capturado em: <http://www.dailymail.co.uk/sciencetech/article-2906394/Google-Translate-app-soon-translate-SPEECH-real-time-insider-reveals.html>, Janeiro 2016.
- [GOO16a] Google. “Your World. Now with 90 Languages”. Capturado em: http://translate.google.com/about/intl/en_ALL/languages.html, Janeiro 2016.
- [GOO16b] Google. “Hallo, hola, olá, to the new, more powerful Google Translate app”. Capturado em <https://googleblog.blogspot.com.br/2015/01/hallo-hola-ola-more-powerful-translate.html>, Janeiro 2016.
- [GOO16c] Google. “Google Translate API”. Capturado em: <https://cloud.google.com/translate/>, Janeiro 2016.
- [HER01] Herbsleb, J. D.; Moitra, D. “Guest Editors’ Introduction: Global Software Development”. *IEEE Software*, vol. 18-2, 2001, pp. 16-20.
- [HER07] Herbsleb, J. D. “Global Software Engineering: The Future of Socio-Technical Coordination”. In: *Proceedings of the 29th International Conference of Software Engineering*, 2007, pp. 188-198.

- [HBR14] Huang, X.; Baker, J.; Reddy, R. "A Historical Perspective of Speech Recognition". *Communications of the ACM*, vol. 57-1, Jan 2014, pp. 94-103.
- [HIS10] História Sobre Sites de Busca. "História do Site de Busca Alta Vista". Capturado em <https://sites.google.com/site/historiasobreos-sitesdebusca/historia-dos-principais-sites-de-busca/historia-do-site-de-busca-alta-vista>, Dezembro 2014.
- [HOL01] Holmes, J.; Holmes, W. "Speech Synthesis and Recognition". *CRC Press*, vol. 2, 2001, pp 91-92.
- [HUT93] Hutchins, J. "The First MT Patents". *MT News International no. 5*, Maio 1993, pp. 14-15.
- [HUT97] Hutchins, J. "From First Conception to First Demonstration: the Nascent Years of Machine Translation, 1947-1954. A Chronology". In: *Machine Translation, Springer*, vol. 12-3, 1997, pp. 195-252.
- [HUT97b] Hutchins, J. "Fifty Years of the Computer and Translation". In *MT News International*, In: *Machine Translation Review no. 6*, Out 1997, pp. 22-24.
- [HUT00] Hutchins, J. "Petr Petrovich Troyanskij (1894-1950): A Forgotten Pioneer of Mechanical Translation". In: *Machine Translation, Springer*, vol. 15-3, 2000, pp. 187-221.
- [HUT04] Hutchins, J. "Two Precursors of Machine Translation: Artsrouni and Troyanskij". In: *International Journal of Translation, Bahri Publications*, vol. 16-1, 2004, pp. 11-31.
- [HUT04b] Hutchins, J. "The Georgetown-IBM Experiment Demonstrated in January 1954". Capturado em http://link.springer.com/chapter/10.1007%2F978-3-540-30194-3_12#, Dezembro 2014.
- [HUT05] Hutchins, J. "The History of Machine Translation in a Nutshell". Capturado em <http://www.hutchinsweb.me.uk/Nutshell-2005.pdf>, Dezembro 2014.
- [ITB10] Brazil IT-BPO Book 2008-2009, publicado por Brasscom Associação Brasileira das Empresas de Tecnologia da Informação e Comunicação, São Paulo, Brasil, 2010, pp. 16-17.

- [JM08] Jurafskyand, D.; Martin, J. H. "Speech and Language Processing 2nd ed.". *Prentice Hall Series in Artificial Intelligence*, Prentice Hall, 2008, pp. 797-798.
- [JUA04] Juang, B. H.; Rabiner, L. R. "Automatic Speech Recognition – A Brief History of the Technology Development". Capturado em: http://www.ece.ucsb.edu/Faculty/Rabiner/ece259/Reprints/354_LA_LI-ASRHistory-final-10-8.pdf, Dezembro 2014.
- [LB09] Lutz, B. "Linguistic Challenges in Global Software Development: Lessons Learned in International SW Development Division". In: *4th IEEE International Conference on Global Software Engineering ICGSE*, 2009, pp. 249-253.
- [LOP05] Lopes, L.; Audy, J. L. N. "Impact of GSD in Requirements Specification – A Case Study". In *Proceedings of the 7th International Conference of Software Engineering and Knowledge Engineering*, 2005, pp. 473-478.
- [LUC16] Alcatel-Lucent. "Text-to-Speech Synthesis". Capturado em: <http://goo.gl/CwdEKI>, Janeiro 2016.
- [MAN10] Mannell, R. "A Brief Historical Introduction to Speech Synthesis: A Macquarie Perspective". Capturado em: http://clas.mq.edu.au/speech/synthesis/history_synthesis/, Dezembro 2014.
- [MIC16a] Microsoft. "Microsoft Speech Platform SDK 11 Requirements and Installation". Capturado em: [https://msdn.microsoft.com/en-us/library/hh362873\(v=office.14\).aspx](https://msdn.microsoft.com/en-us/library/hh362873(v=office.14).aspx), Janeiro 2016.
- [MIC16b] Microsoft. "Microsoft Language Development Center". Capturado em: <https://www.microsoft.com/pt-pt/mldc/downloads.aspx?CollectionId=6628c74c-9625-4f3e-bb43-964a1ff166c3>, Janeiro 2016.
- [MOC01] Mockus, A.; Herbsleb, J. D. "Challenges of Global Software Development". In: *Proceedings of the 7th Software Metrics Symposium*, 2001, pp. 182-184.
- [NUA16] Nuance. "Innovative Technologies Powering our Solutions". Capturado em: <http://www.nuance.com/for-business/customer-service-solutions/speech-technology/index.htm>, Janeiro 2016.

- [NUS00] Nuseibeh, B.; Easterbrook, S. "Requirements Engineering: a Roadmap". *Proc. Int'l Conf on. the Future of Software Engineering (ICSE 00)*, Jun 2000, pp. 35-46.
- [NYT15] New York Times. "Language Translation Tech Starts to Deliver on Its Promise". Capturado em: http://bits.blogs.nytimes.com/2015/01/11/language-translation-tech-starting-to-deliver-on-its-promise/?_r=0, Fevereiro 2016.
- [PFL04] Pfleeger, S. L. "Engenharia de Software: teoria e prática". São Paulo: Person / Prentice Hall, 2004, 2ª edição, 560p.
- [PIN15] Pinto, J. H. S. "Um Estudo Empírico Sobre Máquinas de Tradução em Tempo Real para Equipes Distribuídas de Desenvolvimento de Software". In: *5th Workshop on Thesis and Dissertations of CBSOft*, 2015, pp. 9-14.
- [PM11] Pinola, M. "Speech Recognition Through the Decades: How We Ended Up With Siri". *PCWorld*, Nov 2011, Capturado em: http://www.techhive.com/article/243060/speech_recognition_through_the_decades_how_we_ended_up_with_siri.html?page=2, Dezembro 2014.
- [PRI02] Prikladnicki, R. "Desenvolvimento Distribuído de Software e Processos de Desenvolvimento de Software", Trabalho Individual II, Mestrado em Ciência da Computação, PUCRS, 2002.
- [RED76] Reddy, R. "Speech Recognition by Machine: A Review". *Proceedings of the IEEE*, vol. 64, 1976, pp. 501-531.
- [SIQ05] Siqueira, F. L. "O Desenvolvimento Distribuído de Software: Características e Recomendações para a Gerência de Projetos", Dissertação de Mestrado, Mestrado em Engenharia da Computação, Escola Politécnica da Universidade de São Paulo, 2005.
- [SJ05] Schroeter, J. "Text-to-Speech (TTS) Synthesis". *AT&T Laboratories*, 2005, Capturado em: http://www2.research.att.com/~ttsweb/tts/papers/2005_EEHandbook/tts.pdf, Dezembro 2014.

- [SO99] Sproat, R.; Olive, J. "Text-To-Speech Synthesis". *Digital Signal Processing Handbook*, ed. Vijay K. Madisetti and Douglas B. Williams, Boca Raton: CRC Press LLC, 1999, pp. 976-986.
- [SKY14a] Skype. "Skype Translator – How It Works". Capturado em: <http://blogs.skype.com/2014/12/15/skype-translator-how-it-works/>, Janeiro 2016.
- [SKY14b] Skype. "What is Skype Translator Preview". Capturado em: <https://support.skype.com/en/faq/FA34535/what-is-skype-translator-preview>, Janeiro 2016.
- [SKY15] Skype. "Olá! Skype Translator welcomes Portuguese". Capturado em: <http://blogs.skype.com/2015/12/10/ola-skype-translator-welcomes-portuguese/>, Janeiro 2016.
- [TEC06] Techshout. "Yahoo launches Babel Fish, the Language Translation Service". Capturado em: <http://www.techshout.com/internet/2006/28/yahoo-launches-babel-fish-the-language-translation-service/>, Janeiro 2016.
- [TH00] Traunmüller, H. "Wolfgang von Kempelen's Speaking Machine and its Successors". Capturado em: <http://www2.ling.su.se/staff/hartmut/kemplne.htm>, Dezembro 2014.
- [WAI08] Waibel, A.; Fugen, C. "Spoken language translation". *Signal Processing Magazine, IEEE*, vol. 25-3, Mai 2008, pp. 70-79.
- [WEA49] Weaver, W. "Translation". Capturado em: <http://www.mt-archive.info/50/Weaver-1949-original.pdf>, Dezembro 2014.
- [YH06] Hsieh, Y. "Culture and Shared Understanding in Distributed Requirements Engineering". In: *1st Int'l Conf. on Global Software Engineering (ICGSE '06)*, Out 2006, pp. 101-108.

APÊNDICE A

PROTOCOLO DE ESTUDO - EXPERIMENTO 1

Esse protocolo de estudo foi elaborado em colaboração com os pesquisadores Fabio Calefato e Filippo Lanubile, da Universidade de Bari, na Itália e envolveu alunos estrangeiros e brasileiros. Por esse motivo, o protocolo de estudo foi desenvolvido apenas em inglês.

The Simulation

The overall goal of this preliminary study is to evaluate the feasibility of adopting a real-time speech translation service for supporting cross-language communication in multilingual requirements meetings.

Research questions:

RQ1: *How do errors in the speech recognition process affect the resulting machine translation outcome?*

RQ2: *Which source language works better for non-native English speakers with state-of-the-art speech translation technology?*

RQ3 – *How well does state-of-the-art speech translation technology perform when English is not used as either the source or the target language?*

Unit of Analysis

Utterance: a continuous piece of speech beginning and ending with a clear pause.

Experimental Design

In order to run the simulation, we used the Google Translate mobile app (ver. 3.0.4), with voice input enabled in order to pipe the speech recognition output into the machine translation (see Fig. 1). The tests were executed on the following devices: a Galaxy Note 3, running Android 4.3, at the Italian site; a

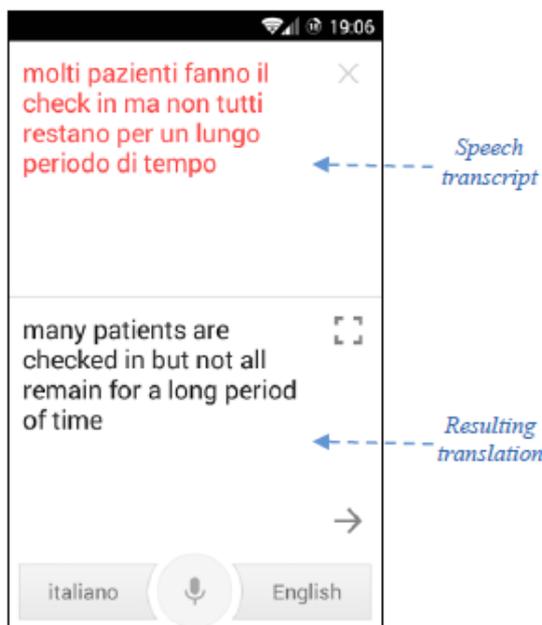
Galaxy Note GT N7000, running Android 4.1.2, at the Brazilian site. Besides, all the tests were executed on the two devices using a Wi-Fi connection.

Preparation

As the test set, we selected 51 sentences of growing length (word count, min. 4, max. 94). The sentences were selected from real chat logs in English, collected from five requirements workshops run as part of an experiment on the effects of text-based communication in distributed requirements engineering.

Instrumentation and Execution

As a speech recognizer we will use the Google Translate mobile app (ver. 3.0.4).



The researcher will speak to the cell phone and the sentence will be recognized by the device. After that, the researcher will have to copy and paste this result into a spreadsheet. The researcher will continue to do that until the sentences are gone. Here follows an example:

UniBari: IT_{IT} → EN ; IT_{IT} → PT

PUCRS: PT_{PT} → EN ; PT_{PT} → IT

The results should be copied and pasted by the researcher into another spreadsheet cell for follow-up rating.

At a pace of two utterances per minute, a speaker will finish in no more than 30 minutes.

Speech Recognition

Independent Variables:

Source **Language:** Italian vs. Portuguese

Operational definition: the language used as a source for speech translation. It should be the native language for participants to the simulation, that is, the language the participant began learning at birth.

Speaker: Sp1 under each language level

Operational definition:

1 speaker, half at UniBari as Italian native speakers and half at PUCRS as Portuguese native speakers.

Dependent Variables

Transcript accuracy: $T_{acc} = \frac{\# \text{ recognized words} - \# \text{ missing words}}{\# \text{ words in utterance}}$

Transcript accuracy can be automatically measured if every transcript is saved after speech recognition. As is, $T_{acc} \in [-1, 1]$. Yet, in order to show values as percentages, the normalized transcript accuracy should vary between 0 and 1. Therefore, we normalize it as follows:

Normalized Transcript accuracy: $T'_{acc} = (T_{acc} + 1)/2$

As per transcription accuracy, our analysis should distinguish between *missing words*, i.e., those that were not recognized at all by the speech recognition component, and *wrong words*, i.e., those that were recognized inaccurately.

Machine Translation

Independent Variables:

Language Pairs: IT → EN; IT → PT; PT → EN; PT → IT;

Operational definition: the combination of source and target language in the automatic machine translation. The utterance in input is the transcript returned by speech recognition.

Lexicon: generic vs. jargon

Operational definition:

A generic utterance contains only words that are included in a source language dictionary.

A jargon utterance contains one or more technical terms or characteristic acronyms used by software developers.

Transcript accuracy: $T_{acc} = \frac{\# \text{ recognized words} - \# \text{ missing words}}{\# \text{ words in utterance}}$

Transcript accuracy is automatically measured for every transcript and saved after speech recognition. As is, $T_{acc} \in [-1, 1]$. Yet, in order to show values as percentages, the normalized transcript accuracy should vary between 0 and 1. Therefore, we normalize it as follows:

Normalized Transcript accuracy: $T'_{acc} = (T_{acc} + 1)/2$

Dependent Variables

Translation adequacy: 4-item Likert scale, anchored with values 4=completely accurate/adequate and 1=completely inaccurate/inadequate. The original utterance is the spoken utterance (not the transcript).

- 4 Completely adequate: the translation clearly reflects the information contained in the original utterance. It is perfectly clear, intelligible, grammatically correct, and reads like ordinary text.
- 3 Fairly adequate: the translation generally reflects the information contained in the original utterance, despite some inaccuracies or infelicities in the text. It is generally clear and intelligible and one can (almost) immediately understand what it means.
- 2 Somewhat adequate: the translation poorly reflects the information contained in the original utterance. It contains grammatical errors and/or poor word choices. The general idea of the text is intelligible only after considerable study.
- 1 Completely inadequate: the translation is unintelligible and it is not possible to obtain the information contained in the original utterance. Studying the meaning of the text is hopeless and, even allowing for context, one feels that guessing would be too unreliable.

Translation adequacy can be measured post-facto by two researchers if all the translations are saved. The inter-rater agreement will be measured by means of the Spearman's rank correlation coefficient.

APÊNDICE B

PROTOCOLO DE ESTUDO - EXPERIMENTO 2

Esse protocolo de estudo foi elaborado em colaboração com os pesquisadores Fabio Calefato e Filippo Lanubile, da Universidade de Bari, na Itália e envolveu alunos estrangeiros e brasileiros. Por esse motivo, o protocolo de estudo foi desenvolvido apenas em inglês.

The Simulation

The overall goal of this simulation is to assess the usage of real-time speech translation to support generation of content in multilingual requirements meetings.

Research questions:

RQ1: *how does speaking without interruptions affect speech translation?*

RQ2: *how does technical jargon affect speech translation?*

Unit of Analysis

Utterance: a continuous piece of speech beginning and ending with a clear pause.

Experimental Design

Subjects will be recruited as speakers. Information about gender, age, instruction and years of work experience will be recorded. At the initial stage, there will be 4 professional software engineers as speakers for each side at UniBari and PUCRS where UniBari and PUCRS will use respectively Italian and Portuguese as source language.

Subjects (i.e., speakers) in each of 2 groups at UniBari and PUCRS will be exposed to speech tasks under 2 different levels of lexicon in the utterance to be spoken, that is generic and jargon utterances. Each subject will perform each task 30 times under each lexicon level, for a total of 60 utterances to be spoken per subject.

In order to avoid biases, utterances will be spoken alternating one generic utterance and one jargon utterance.

Preparation

The original utterances to be used for simulation are in English and come from the chat logs of real requirements meetings. One the researcher will select 30 utterances for each level of the lexicon independent variable, for a total of 60 sentences. The length of the original utterances will vary between 5 and 30 words and the distribution will be equal between generic and jargon utterances.

Then utterances should be manually translated by the researchers in both Italian and Portuguese. The set of original utterances in English together with the manual translation in Italian and Portuguese will make the experimental sample.

Instrumentation and Execution

As a speech recognizer we will use the Google demo version of the Web Speech API available for the Chrome browser at the URI: <https://www.google.com/intl/en/chrome/demos/speech.html>

For each of the 60 utterances in the sample, a speaker should start by clicking on the microphone icon and begin speaking until the end of the utterance. Participants should speak in a colloquial style at their own pace. If the researcher realizes that the spoken utterance is different from what is written or the speaker stops before arriving to the end of the utterance, then the researcher will invite the speaker to try again.

Once the transcript appears in the text field it should be copied and pasted by the researcher into a sheet, named as the speaker, of the spreadsheet. *Data collection form simulation IT.xls* and *Data collection form simulation PT.xls*,

respectively for UniBari and PUCRS. After all the sixty utterances have been spoken, the researcher can save and close the spreadsheet, and the speaker can leave the simulation session.

After speech recognition has been completed, one of the researchers at each side will perform machine translation on the respective transcripts recorded in the spreadsheets. Each transcript should be copied and pasted into the input text field of the Google Web Translator at the URI <http://translate.google.com> to be translated from the source language (Italian or Portuguese) into both English and the native language of the other side: either Italian or Portuguese. The following summarizes translations¹:

UniBari: $IT_{IT} \rightarrow EN$; $IT_{IT} \rightarrow PT$

PUCRS: $PT_{PT} \rightarrow EN$; $PT_{PT} \rightarrow IT$

The results should be copied and pasted by the researcher into another spreadsheet cell for follow-up rating.

At a pace of two utterances per minute, a speaker will finish in no more than 30 minutes.

Pilot Simulation

In order to avoid waste of time, one of researchers from both sides will act as speaker on a subset of 20 utterances. Results from the pilot will be assessed to make changes. If results are not positive or after making changes, a second pilot simulation will be performed by including the other researcher as a speaker. Thereafter the researchers will decide whether the simulation is ready for recruiting software engineering professionals as speakers. Speech Recognition Independent Variables:

Source Language: Italian vs. Portuguese

¹ The following example explains the notation. $IT_{(IT)} \rightarrow EN$ means that an utterance in Italian is read in by an Italian native speaker and translated into English.

Operational definition: the language used as a source for speech translation. It should be the native language for participants to the simulation, that is, the language the participant began learning at birth.

Speaker: Sp1-Sp4 under each language level

Operational definition:

8 speakers, half at UniBari as Italian native speakers and half at PUCRS as Portuguese native speakers.

Lexicon: generic vs. jargon

Operational definition:

A generic utterance contains only words that are included in a source language dictionary.

A jargon utterance contains one or more technical terms or characteristic acronyms used by software developers.

Replication: Rep1-Rep30 under each lexicon level

Operational definition:

Speech recognition tasks are repeated 30 times under each lexicon level, for a total of 60 utterances to be spoken per subject.

Lexicon is a fixed effect factor while (source) language is a random fixed factor. The 30 replications under each lexicon level are considered a random factor nested under lexicon. Speakers are nested within languages.

Dependent Variables

Transcript accuracy: $T_{acc} = \frac{\# \text{ recognized words} - \# \text{ missing words}}{\# \text{ words in utterance}}$

Transcript accuracy can be automatically measured if every transcript is saved after speech recognition. As is, $T_{acc} \in [-1, 1]$. Yet, in order to show values as percentages, the normalized transcript accuracy should vary between 0 and 1. Therefore, we normalize it as follows:

Normalized Transcript accuracy: $T'_{acc} = (T_{acc} + 1)/2$

As per transcription accuracy, our analysis should distinguish between *missing words*, i.e., those that were not recognized at all by the speech recognition component, and *wrong words*, i.e., those that were recognized inaccurately.

Machine Translation

Independent Variables:

Language Pairs: IT → EN; IT → PT; PT → EN; PT → IT;

Operational definition: the combination of source and target language in the automatic machine translation. The utterance in input is the transcript returned by speech recognition.

Lexicon: generic vs. jargon

Operational definition:

A generic utterance contains only words that are included in a source language dictionary.

A jargon utterance contains one or more technical terms or characteristic acronyms used by software developers.

Transcript accuracy: $T_{acc} = \frac{\# \text{ recognized words} - \# \text{ missing words}}{\# \text{ words in utterance}}$

Transcript accuracy is automatically measured for every transcript and saved after speech recognition. As is, $T_{acc} \in [-1, 1]$. Yet, in order to show values as

percentages, the normalized transcript accuracy should vary between 0 and 1. Therefore, we normalize it as follows:

Normalized Transcript accuracy: $T'_{acc} = (T_{acc} + 1)/2$

Dependent Variables

Translation adequacy: 4-item Likert scale, anchored with values 4=completely accurate/adequate and 1=completely inaccurate/inadequate. The original utterance is the spoken utterance (not the transcript).

- 4 Completely adequate: the translation clearly reflects the information contained in the original utterance. It is perfectly clear, intelligible, grammatically correct, and reads like ordinary text.
- 3 Fairly adequate: the translation generally reflects the information contained in the original utterance, despite some inaccuracies or infelicities in the text. It is generally clear and intelligible and one can (almost) immediately understand what it means.
- 2 Somewhat adequate: the translation poorly reflects the information contained in the original utterance. It contains grammatical errors and/or poor word choices. The general idea of the text is intelligible only after considerable study.
- 1 Completely inadequate: the translation is unintelligible and it is not possible to obtain the information contained in the original utterance. Studying the meaning of the text is hopeless and, even allowing for context, one feels that guessing would be too unreliable.

Translation adequacy can be measured post-facto by two researchers if all the translations are saved. The inter-rater agreement will be measured by means of the Spearman's rank correlation coefficient.