Unbabel generates millions of translations of customer support tickets and chats every month. Each piece of text, such as a message in a chat, is translated with machine translation (MT) and then checked by a human and post-edited if necessary. In order to maintain the quality of the machine translation, Unbabel relies on a community of human annotators to annotate errors in the translations. However, to have humans look at everything is very expensive and very slow. As such, Unbabel also relies on automated metrics for MT to make development and deployment decisions about their MT models.

Recently Unbabel developed COMET, a new framework for evaluating MT models. To decide whether or not automated metrics such as COMET can do this job effectively, MT researchers usually take samples of human annotated translation and look at the correlation between automated metric scores and human scores of quality. However, looking at correlations doesn't give researchers much of an idea about what kinds of error the metrics can catch and whether the most critical errors are being effectively highlighted in metric scores. My summer internship at Unbabel was focused on developing a way to test COMET and other automated metrics on specific kinds of errors.

I learned a lot about how technology companies work during this internship. My first weeks at Unbabel were a little bit intimidating. The company culture was very fast-paced, and there were meetings constantly. Additionally, Unbabel used a software/product development workflow called "agile" development. In agile development, teams work in short-term "sprints," which are usually two weeks long. At the beginning of each sprint, my team got together to plan out all the tasks to be completed in the next two weeks. We also estimated how many hours each task would take. Then we input all the tasks with the estimated times into our work management system, Jira. As we made progress in the sprint, we updated the tasks in Jira. At the end of the

two weeks, we had a retrospective meeting to talk about what tasks were completed and if anyone ran into any problems. This kind of workflow was very different from the academic world where it feels like people can work on their own for long periods of time. However, I really enjoyed working in this very structured workflow and knowing exactly what I needed to do in each sprint.

I also learned a lot about translation quality evaluation during my internship. Unbabel has its own internal annotation guide for its community of human annotators. The guide is based on the commonly-used Multidimensional Quality Metrics (MQM). MQM defines over 100 different issues that may show up in translations. These issues are arranged in a hierarchy of categories. At the top level, there are eight major categories or "dimensions": Accuracy, Fluency, Terminology, Locale convention, Style, Verity, Design, and Internationalization. Unbabel's annotation guide uses three of these dimensions: Accuracy, Fluency, and Style. I needed to learn all the issues in each of these dimensions and be able to identify them in real translations. To practice identifying these issues, the Team Lead of Linguistic Services gave me a bunch of English-to-Chinese machine translations to annotate. The annotation process was extremely tedious but really helped me learn Unbabel's annotation guide.

After I became familiar with the annotation guide, I began to work on the meat of my internship project—developing a program that tests various automatic metrics on different kinds of errors. I faced a few challenges while creating this program. First, I needed a large set of translations on which to test the program. Additionally, this set of translations needed to have two different target translations for the same source text. This way, I could introduce errors into one of the translations and then compare the scores that a metric gave to each translation. I was able to find a large set of Chinese-to-English and Russian-to-English translations of news articles

that fit all of my criteria. The second challenge was to find a way to create different kinds of errors in a text without doing it by hand. Thankfully, there were a couple existing computer programs that could already do this for simple mistakes, such as incorrect spelling, punctuation, or numbers. All I had to do was integrate these programs into the program that I was writing.

The actual process of writing the program for testing metrics was fairly straightforward. I wrote one piece of code that took the new article translations and introduced errors into one of the translations for each article. Then I wrote another piece of code that took a metric, had it evaluate the news article translations, and compared the scores of the error-free and error-full translations. Once I finished writing the program, I tested four versions of COMET developed by Unbabel and four other evaluation metrics. The results showed that COMET performed very well compared to other metrics but that it had trouble with a couple types of errors. This information was very useful to the developers of COMET, since they were about to submit a new version to the Sixth Conference on Machine Translation.

Overall, I was very pleased with my internship at Unbabel. It was my first time working at a tech company, and experiencing that kind of work environment was invaluable. I was also extremely happy to be able to produce something concrete at the end of the internship that was helpful to other people in the company. My experience at Unbabel will definitely help me in the future if I decide to continue a career in the language technologies industry.