# Creating translation context with disambiguation in ITS2.0

|  |  |
| --- | --- |
| Text | Scene |
| When you mention names in your content, you are faced with specific rules to translate (or transliterate) proper names. Sometimes, they should not even be translated. The correct course of action often depends on the context.  | Intro slide, illustration. |
| For example, when you say: “I’m moving to London,” I’d usually interpret that as you’re moving to the UK.  | Show example “I’m moving to London” |
| However, if you continue that sentence with “I hope it doesn’t get too cold in Ontario”. With the added context, the meaning of London switches to another city - London in Ontario. | Show example “I’m moving to London. I hope it doesn’t get too cold in Ontario.” |
| In order to support this use case, you can use natural processing tools to annotate the content so that it can be correctly processed by translators and MT systems. |  |
| Here I’d like to show you how you can use text analysis tools to provide the necessary markup. | Open enrycher |
| This is Enrycher. It’s a web service that receives textual content, and returns it marked-up with identities of named entities that it finds in your text. | Enter content, click ‘enrych’ |
| What you see here in the output are the necessary annotations that were produces automatically. It’s valid HTML5, extended by ITS2.0 metadata.  | Show result.Show ITS2.0 sketch |
| The ITS2.0 properties give you the necessary information to handle this situation: what is the identity of London, what kind of entity is “London”, and what tool produced all this markup. | Show DBpedia page for London, Ontaro. |
| The great part about ITS2.0 being a W3C recommendation is that this kind of markup is understood by several implementations, so that integrating text analysis tools into your localization workflow is not easier than ever. | Outro |