Techniques:

- Speaker dependent
  Engine needs to be trained by the user, so the engine can decode the spoken text more accurately.
- Cloud based big data
  The engine is trained with lots of data from the cloud. Therefore the engine has to be connected to the internet the whole time.
- GPU Accelerated Deep Learning in Deep Neural Networks
  A bit different than Cloud based big data. The engine train itself independently with big data using virtual neural networks.

Researched Speech Recognition Engines:

Engine	**Specifications**	**Advantages**	**Disadvantages**
Open Source		Free	
CMU Sphinx4	Last release: 3-8-2015, Java, BSD style license	Language learning, Crossplatform, Great documentation, Forum support	No neural network support.
CMU Pocketsphinx	C, BSD style license	Faster than Sphinx4 (real time), Less memory usage, Less storage space needed(50MB)	Less accurate
Kaldi	Last release: October 2013, C++, Apache License v2	Crossplatform, Supports multiple algorithms, Deep neural support, Github	Documentation is for SR experts
Julius	Last release: January 2014, C, BSD style license		No OSX support
Commercial			Online required
Dragon Nuance	C, C++, Visual Basic	Most accurate engine 2015, Used by Apple’s Siri, Great support center, Easy to use	Expensive
Google’s SR API	Javascript, Audio in FLAC/SPEEX format	Only commercial cloud based engine with a free API, Speaker independent	No language module configuration, No vocabulaire configuration, API can be closed any time*1
NAO’s SR	Any supported language	Uses Nuance Dragon’s technique for SR, Built in mic NAO, Easy testing in Choregraphe	Speaking frontal to the robot is less accurate, Can customize options but not train

*1 This happened with the Google Translate API.

Conclusion:
The best option from my research would be Baidu's Deep Speech. This SR Engine use neural networks based on deep learning and is specialized in noisy environments. This option would be perfect to use in an elementary school full of screaming kids. The technique is unfortunately still very new and there are no available API's.
Kaldi's SR Engine looks very promising, because it supports different algorithms. One of those supported algorithms is neural networks, a technique of the future. But by taking a look at the documentation, I can see that it's made for experts in this field. Therefore I do not think I am capable of using this engine.
The best option for now would be using CMU's Sphinx4 or pocketsphinx. Pocketsphinx is more lightweight and reacts real time, which makes it faster than Sphinx4 but less accurate.
Another great option to consider is to use the SR Engine on the NAO robot itself. I tested it quickly and it seems to work very well from the sides. Frontal speaking to the robot was not very accurate.